PATENT APPLICATION

#4 Priority Paper 4-4-00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of

Shinichiro TANIGUCHI et al. Group Art Unit: 2734

Application No.: 09/454,865

Filed: December 7, 1999 Docket No.: 104934

For: DISTRIBUTION INFORMATION MANAGEMENT SYSTEM AND METHOD

CLAIM FOR PRIORITY

Assistant Commissioner for Patents
Washington, D.C. 20231.

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. 11-115551 filed April 22, 1999

In support of this claim, a certified copy of said original foreign application:

__X__ is filed herewith.

_____ was filed on _____ in Parent Application No. _____ filed _____.

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of this document.

Respectfully submitted,

James A. Oliff
Registration No. 27,075

Thomas J. Pardini
Registration No. 30,411

JAO:TJP/kmc

DEPOSIT ACCOUNT USE AUTHORIZATION
Please grant any extension necessary for entry;
Charge any fee due to our Deposit Account No. 15-0461

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年　４月２２日

出願番号
Application Number: 平成１１年特許願第１１５５５１号

出願人
Applicant (s): 富士ゼロックス株式会社

RECEIVED
FEB 17 2000
Group 2700

１９９９年１０月２９日

特許庁長官
Commissioner,
Patent Office 近藤隆彦

出証番号　出証特平１１－３０７４８２５

【書類名】　特許願

【整理番号】　FN99-00050

【提出日】　平成11年 4月22日

【あて先】　特許庁長官殿

【国際特許分類】　G09G 3/00

【発明の名称】　物流情報管理装置および方法

【請求項の数】　27

【発明者】

【住所又は居所】　神奈川県足柄上郡中井町境４３０　グリーンテクなかい
富士ゼロックス株式会社内

【氏名】　谷口　慎一郎

【発明者】

【住所又は居所】　神奈川県足柄上郡中井町境４３０　グリーンテクなかい
富士ゼロックス株式会社内

【氏名】　河野　健二

【発明者】

【住所又は居所】　神奈川県足柄上郡中井町境４３０　グリーンテクなかい
富士ゼロックス株式会社内

【氏名】　申　吉浩

【特許出願人】

【識別番号】　000005496

【氏名又は名称】　富士ゼロックス株式会社

【電話番号】　0462-38-8516

【代理人】

【識別番号】　100086531

【弁理士】

【氏名又は名称】　澤田　俊夫

【電話番号】　03-5541-7577

【手数料の表示】

【予納台帳番号】　038818

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　流通情報管理装置および方法

【特許請求の範囲】

【請求項１】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と、

上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、

上記物流情報処理モジュールとの通信を行うための第二の通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための第二の情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記第二の情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための第二の情報検証手段と、

上記第二の情報検証手段が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段と、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理装置。

【請求項２】　上記署名モジュールは、上記物流情報処理モジュールから取り外し交換することができることを特徴とする請求項１の物流情報管理装置。

【請求項３】　上記署名モジュールは、耐タンパー性を持つことを特徴とする請求項１または２に記載の物流情報管理装置。

【請求項４】　上記情報生成部は、署名鍵使用限度情報保持手段を持ち、上記署名鍵情報選択手段は、特定回数以上署名に用いた署名鍵情報を選択しないことを特徴とする請求項１、２または３に記載の物流情報管理装置。

【請求項５】　上記署名鍵使用限度情報保持手段は、上記署名モジュール内にあることを特徴とする請求項１、２、３または４に記載の物流情報管理装置。

【請求項６】　上記物流情報処理モジュールは、

情報検証モジュールと、

情報生成モジュールと、

から構成され、

上記情報検証モジュールは、

上記データキャリアのデータを読み出す第一の読み取り手段と、

上記データキャリアから読み出した情報を処理する第一の情報検証部と、

から構成され、

上記情報生成モジュールは、

上記データキャリアのデータを読み出す第二の読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアへ書込む情報を処理する情報生成部と、

から構成される

ことを特徴とする請求項１、２、３、４または５に記載の物流情報管理装置。

【請求項７】　上記物流情報管理モジュールは、

第二の情報検証モジュールと、

署名鍵情報生成モジュールと、

から構成され、

上記第二の情報検証モジュールは、

物流情報検証部と、

第二の通信手段と、

から構成され、

上記署名鍵情報生成モジュールは、

署名鍵生成部と、

第三の通信手段と、

から構成される

ことを特徴とする請求項１、２、３、４、５または６に記載の物流情報管理装置。

【請求項８】　上記第一の検証鍵記憶手段と上記第二の検証鍵記憶手段に記憶される検証鍵は、全ての物流情報処理モジュールと物流情報管理モジュールとで共通であることを特徴とする請求項１～７のいずれかに記載の物流情報管理装置。

【請求項９】　第一の情報検証手段と第二の情報検証手段は、同じ動作をすることを特徴とする請求項１～８のいずれかに記載の物流情報管理装置。

【請求項１０】　第一の情報検証部は、第一の情報検証手段が使用する検証鍵を選択する第一の検証鍵選択手段を持つことを特徴とする請求項１～８のいずれかに記載の物流情報管理装置。

【請求項１１】　第二の情報検証部は、第二の情報検証手段が使用する検証鍵を選択する第二の検証鍵選択手段を持つことを特徴とする請求項９または１０に記載の物流情報管理装置。

【請求項１２】　署名鍵情報生成部は、署名鍵を選択する署名鍵選択手段を持つことを特徴とする請求項１～１１のいずれかに記載の物流情報管理装置。

【請求項１３】　データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されることを特徴とする請求項１～１２のいずれかに記載の物流情報管理装置。

【請求項１４】　データキャリアに格納される情報は少なくとも、検証鍵識別子とを含む情報を一単位として格納されることを特徴とする請求項１３の物流

情報管理装置。

【請求項１５】　データキャリアに格納される情報は少なくとも、物流情報管理モジュール識別子とを含む情報を一単位として格納されることを特徴とする請求項１１、１２、１３、または１４に記載の物流情報管理装置。

【請求項１６】　データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つことを特徴とする請求項１～１２のいずれかに記載の物流情報管理装置。

【請求項１７】　データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つことを特徴とする請求項１～１２のいずれかに記載の物流情報管理装置。

【請求項１８】　物品に添付され、上記物品に関する情報を記憶するデータキャリアにおいて、

上記物品の流通過程における１またはひとまとまりの取引ごとに生成される上記物品の流通情報と、１の上記流通情報の少なくとも一部または一連の上記流通情報のおのおのの少なくとも一部の署名値とを記憶することを特徴とするデータキャリア。

【請求項１９】　上記物品の流通情報は、上記物品の識別子と、上記物品を受領した者の識別子と、上記署名値を生成する者の識別子とを少なくとも含む請求項１８記載のデータキャリア。

【請求項２０】　物品に添付され、上記物品に関する情報を記憶するデータキャリアに情報を読み書きするとともに、上記物品の流通に関する情報を管理する物流情報管理モジュールとの間で情報をやり取りして上記物品に関する情報を処理する物流情報処理モジュールにおいて、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と、

上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成される、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成されること特徴とする物流情報処理モジュール。

【請求項２１】　物品に添付され、上記物品に関する情報を記憶するデータキャリアに情報を読み書きするとともに上記データキャリアの情報を検証するための物流情報処理モジュールとの間で情報をやり取りし、上記物品の流通に関する情報を管理する物流情報管理モジュールにおいて、

上記物流情報処理モジュールとの通信を行うための通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証手段と、

上記情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理モジュール。

【請求項２２】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理する方法において、

上記データキャリアのデータを読み出す読み取りステップと、

上記データキャリアに情報を書込む書き込みステップと、

上記データキャリアから読み出した情報を検証する第一の情報検証ステップと、

上記データキャリアへ書込む情報を処理する情報生成ステップと、

上記物流情報管理モジュールとの通信を行うための第一の通信ステップと、

から構成され、

上記第一の情報検証ステップは、

上記データキャリアから読み出した情報の検証を行う第一の情報検証サブステップと、

上記第一の情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶サブステップと、

から構成され、

上記情報生成ステップは、

上記データキャリアに書込む情報を生成する物流情報生成サブステップと、

署名生成処理を行う署名サブステップと、

上記署名サブステップにおいて電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶ステップと、

上記署名鍵記憶サブステップにおいて記憶された署名鍵情報を選択するための署名鍵情報選択ステップと、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得サブステップと、

から構成され、

上記署名サブステップは、

上記物流情報生成モジュールによって生成された情報に対する電子署名を生成する署名マイクロステップと、

上記署名マイクロステップにおいて電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶マイクロステップと、

から構成されることを特徴とする物流情報処理方法。

【請求項２３】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理する方法において、

上記物流情報処理モジュールとの通信を行うための通信ステップと、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証ステップ部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成ステップと、

から構成され、

上記情報検証ステップは、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証サブステップと、

上記情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する検証鍵記憶サブステップと、

から構成され、

上記署名鍵情報生成ステップは、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択サブステップと、

上記署名者秘密情報を記憶する署名者秘密情報記憶サブステップと、

から構成される

ことを特徴とする物流情報管理方法。

【請求項２４】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理するために用いられる物流情報処理用のコンピュータプログラム製品において、

上記データキャリアのデータを読み出す読み取りステップと、

上記データキャリアに情報を書込む書き込みステップと、

上記データキャリアから読み出した情報を検証する第一の情報検証ステップと、

上記データキャリアへ書込む情報を処理する情報生成ステップと、

上記物流情報管理モジュールとの通信を行うための第一の通信ステップと、

をコンピュータシステムに実行させるために用いられ、

上記第一の情報検証ステップは、

上記データキャリアから読み出した情報の検証を行う第一の情報検証サブステップと、

上記第一の情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶サブステップと、

から構成され、

上記情報生成ステップは、

上記データキャリアに書込む情報を生成する物流情報生成サブステップと、

署名生成処理を行う署名サブステップと、

上記署名サブステップにおいて電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶ステップと、

上記署名鍵記憶サブステップにおいて記憶された署名鍵情報を選択するための署名鍵情報選択ステップと、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得サブステップと、

から構成され、

上記署名サブステップは、

上記物流情報生成モジュールによって生成された情報に対する電子署名を生成する署名マイクロステップと、

上記署名マイクロステップにおいて電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶マイクロステップと、

から構成されることを特徴とする物流情報処理用コンピュータプログラム製品。

【請求項２５】　物品に添付され、上記物品に関する情報を記憶するデータ

キャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理するために用いられるコンピュータプログラム製品において、

上記物流情報処理モジュールとの通信を行うための通信ステップと、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証ステップ部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成ステップと、

をコンピュータシステムに実行させるために用いられ、さらに、

上記情報検証ステップは、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証サブステップと、

上記情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する検証鍵記憶サブステップと、

から構成され、

上記署名鍵情報生成ステップは、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択サブステップと、

上記署名者秘密情報を記憶する署名者秘密情報記憶サブステップと、

から構成される

ことを特徴とする物流情報管理用コンピュータプログラム製品。

【請求項２６】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを

含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う情報検証手段と、

上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、

上記物流情報処理モジュールとの通信を行うための第二の通信手段と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成モジュールが署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成モジュールが署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理装置。

【請求項２７】　物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、

上記物流情報処理モジュールとの通信を行うための第二の通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証手段と、

上記情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、物品にデータキャリアを添付し、データキャリア内の情報を利用して物品の流通を管理する技術に関する。

【０００２】

【従来の技術】

従来の流通管理システムとしては、商品の管理に電磁的に読み書き可能なデータキャリアを使用することで、流通過程での作業の効率や確実性を向上させようとするものがある。

【０００３】

例えば、特開平５－２９８３３２号公報の「データキャリアを用いた物流システム」には、非接触で情報の読み書きが可能なデータキャリアを用いた流通システムが記述されている。データキャリアには商品固有のデータが書き込まれており、例えば、品名、産地、個数などが記載されている。商品が競り落とされた場合、データキャリアには仲買人のＩＤが追加記載される。仲買人が商品を小売りに売る場合はさらに小売り人のＩＤ、個数、値段、発送日時などが追加記載される。また、これら商品を乗せる台車やトラックにもデータキャリアが設けられており、積み荷の個数など台車に積まれた商品全体の情報が予め書かれている。商品は小売り人のＩＤにより自動的にそれぞれの台車に振り分けられ、さらに同ＩＤの記載されたトラックに積み込まれる。

【０００４】

このような方法を用いることで、商品の発送先、発送日時、発送個数などの間違いを回避することができる。更にまたデータキャリアを市場への出入りチェッ

クに用いることで商品の盗難防止を図ることができる。

【０００５】

また、特開平１０－３２４４０５号公報の「商品配送システム」には、電子荷札により商品の内容物の確認を正確に行うことができる商品配送システムが記述されている。商品には予め電子荷札が取り付けられており、配送側において、電子荷札から無線により商品情報を読み取って、梱包物内の商品を示す内容物情報を作成し、受け取り側に渡す。受け取り側では、同様に電子荷札から無線により商品情報を受け取って、内容物情報と比較する事で検品を行う。

【０００６】

このような商品配送システムを用いれば、各商品に取り付けられた電子荷札により梱包後に内容物の確認を行い、また受け取り側において梱包した状態で電子荷札から読み取った商品情報と商品のリストを示す内容物情報とを比較して検品を行うので、人間を介在させずに検品を行う事ができ、商品の配送およびその確認を正確に行う事ができる。

【０００７】

これら流通管理システムの改良により商品配送管理の正確性が向上してきたが、今なお存在する問題として、ブランド品などにおいて商品の流通過程で偽物が混入するという問題がある。上記に述べた方法によれば、確かに配送作業を自動化でき商品の検品を正確に行う事ができるが、流通業者が不正を行うことでその流通過程に偽物を混入させることは容易であり、また偽物が混入した場合にどの流通業者が混入したのかをつきとめることは非常に困難であった。

【０００８】

このような不正を排除するために、上記電子荷札（データキャリア）に情報を書き込む際に署名を付けて書き込むという改善策が考えられる。以後の説明では電子荷札をデータキャリアと呼びかえる。データキャリアには各流通業者の取り扱い記録とそれに対する署名が書き込まれているので、偽物が発見された場合は、流通経路を特定することで偽物を混入した業者をある程度特定することが可能である。また、流通業者の取り扱い情報の署名が付加されていなかったり不正だった場合は、その業者またはそこから商品を受け取った業者が怪しいということ

になる。

【０００９】

このような流通情報に対して署名を付ける技術としては、ディジタル署名技術が挙げられる。ディジタル署名技術の代表的な例として、離散対数問題の困難性に安全性の根拠をおくエルガマル署名法がある。以下、エルガマル署名法について説明する。

【００１０】

署名鍵を（ｘ，α，ｐ）、検証鍵を（ｙ，α，ｐ）とする。ここでｐは素数、αはｐ未満のある正定数である。これらの整数の間には、式（１）で示す関係がある。

【００１１】

【数１】

（１）　$y=\alpha^{x} \pmod{p}$

公開整数ｙから秘密整数ｘを求めるのは離散対数問題であり、ｐが十分大きければ（５００ビット以上）ｘを計算により求めることは困難である。

【００１２】

証明者はｐ－１と互いに素な乱数ｋを生成し、式（２）（３）により、メッセージｍに対する署名（ｒ，ｓ）を計算する。

【００１３】

【数２】

（２）　$r=\alpha^{k} \pmod{p}$

【００１４】

【数３】

（３）　$s=(h(m)-xr)\,k^{-1} \pmod{p-1}$

ここで、ｈは一方向性ハッシュ関数を表す。証明者はメッセージｍと署名（ｒ，ｓ）を検証側に送る。

【００１５】

検証者はｍ，（ｒ，ｓ）を受け取って、式（４）が成立するかを確認する。

【００１６】

【数４】

（４）　　$\alpha^{h(m)} = y^{r} r^{s} \pmod{p}$

式（４）が成立することで、ｍが証明者により作成されたメッセージであることが証明される。

【００１７】

ディジタル署名法にはこの他に、同様に離散対数問題の困難さに安全性の根拠を置くＤＳＡ（Ｄｉｇｉｔａｌ　Ｓｉｇｎａｔｕｒｅ　Ａｌｇｏｒｉｔｈｍ）、Ｓｃｈｎｏｒｒ署名法や、零知識証明法に基づいたＧ－Ｑ（Ｇｕｉｌｌｏｕ，Ｑｕｉｓｑｕａｔｅｒ）および、一般的によく知られるＲＳＡ（Ｒｉｖｅｓｔ，Ｓｈａｍｉｒ，Ａｄｌｅｍａｎ）署名法等が挙げられる。

【００１８】

【発明が解決しようとする課題】

しかし、以上で説明したようなディジタル署名法を、上記データキャリアに書き込む流通情報などの署名に用いた場合は、次のような課題がある。

【００１９】

ディジタル署名法を用いる場合は、流通業者各々の署名鍵に対する検証鍵の証明書（Ｃｅｒｔｉｆｉｃａｔｅ）を発行するための認証局（Ｃｅｒｔｉｆｉｃａｔｅ　Ａｕｔｈｏｒｉｔｙ）が必要となる。このような認証局を全国または世界中の流通業者をサポートできるような規模にしようとすると膨大な設備と費用がかかるという問題がある。また、流通情報に対する署名を検証する場合、検証者は署名をした流通業者各々に対して、証明書が必要になり、検証者は認証を行う証明者に対する証明書をその都度認証局から入手するか、予めテーブルに保管しておく必要がある。証明書をその都度入手する方法は、署名者が複数に及ぶ場合は検証に時間がかかるという問題があり、テーブルに保管する場合は、証明書の有効期限の管理や無効化された場合のチェック等の複雑な管理が必要となる。また署名鍵は機密情報なのでその管理にセキュリティ上の注意が必要となる。このようにディジタル署名法を用いると、認証局が必要となり、また証明書の管理や流通業者でのセキュリティの確保等、多大な設備や手間がかかるという問題がある。

【００２０】

また、一般的にディジタル署名を利用する場合は、署名鍵は署名者が管理するので署名回数を制限することができない。その場合、同一商品に対して複数回の署名を施すことが可能であり、署名を行う流通業者が偽造商品に本物の商品と同一のＩＤを振り、別の受け取り業者へ配送するということが可能である。このとき商品に付けられた署名は検証に成功するので、同一のＩＤを持つ商品の存在が発覚するまで偽造商品が本物として出回ることとなってしまうという問題がある。

【００２１】

本発明は、上述の事情を考慮してなされたものであり、多大な設備や手間を必要とすることなく、署名を行え、しかも、署名の回数を制限することができる物品流通管理技術を提供することを目的としている。

【００２２】

【課題を解決するための手段】

本発明の物流情報管理装置は、上述の目的を達成するために、物品に添付され、物品に関する情報を記憶するデータキャリアと；データキャリアの情報を読み書きするための物流情報処理モジュールと；物品の流通に関する情報を管理する物流情報管理モジュールとから構成される。

【００２３】

上記物流情報処理モジュールは：上記データキャリアのデータを読み出す読み取り手段と；上記データキャリアに情報を書込む書き込み手段と；上記データキャリアから読み出した情報を検証する第一の情報検証部と；上記データキャリアへ書込む情報を処理する情報生成部と；上記物流情報管理モジュールとの通信を行うための第一の通信手段とから構成されている。

【００２４】

そして上記第一の情報検証部は：上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と；上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段とから構成されている。

【００２５】

また、上記情報生成部は：上記データキャリアに書込む情報を生成する物流情報生成手段と；署名生成処理を行う署名モジュールと；上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と；上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と；上記署名鍵情報を上記物流情報管理モジュールから取得するための署名鍵情報取得手段とから構成されている。

【００２６】

上記署名モジュールは：上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と；上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段とから構成されている。

【００２７】

また、上記物流情報管理モジュールは：上記物流情報処理モジュールとの通信を行うための第二の通信手段と；上記物流情報処理モジュールから送られてきた情報を処理するための第二の情報検証部と；上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部とから構成されている。

【００２８】

上記第二の情報検証部は：上記物流情報処理モジュールから送られてきた情報を検証するための第二の情報検証手段と；上記第二の情報検証手段が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段とから構成されている。

【００２９】

上記署名鍵情報生成部は：上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と；上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と；上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と；上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段とから構成されている。

【００３０】

このような構成によれば、物流情報処理モジュールは、物流情報管理モジュールからモジュール固有の署名鍵情報を取得し、この署名鍵情報と署名書秘密情報

とを用いて物流情報に署名を行い、この署名の検証には通有性のある検証鍵を用いるようになっている。したがって、署名者ごとの検証鍵をやり取りする必要がなく、設備や手間を少なくすることができる。

【００３１】

また、本発明の物流情報管理装置において、署名モジュールは、物流情報処理モジュールから取り外し交換することができるようにしてもよい。

【００３２】

また、本発明の物流情報管理装置において、署名モジュールは、耐タンパー性を持つようにしてよい。

【００３３】

また、本発明の物流情報管理装置において、情報生成部は、署名鍵使用限度情報保持手段を持ち、署名鍵情報選択手段は、特定回数以上署名に用いた署名鍵情報は選択されないようにしてもよい。

【００３４】

また、本発明の物流情報管理装置において、署名鍵使用限度情報保持手段は、署名モジュール内に存在するようにしてもよい。

【００３５】

また、本発明の物流情報管理装置において、物流情報処理モジュールは、情報検証モジュールと、情報生成モジュールとから構成され、情報検証モジュールは、データキャリアのデータを読み出す第一の読み出し手段と、データキャリアから読み出した情報を処理する第一の情報検証部とから構成され、情報生成モジュールは、データキャリアのデータを読み出す第二の読み出し手段と、データキャリアに情報を書込む書き込み手段と、データキャリアへ書込む情報を処理する情報生成部とから構成されるようにしてもよい。

【００３６】

また、本発明の物流情報管理装置において、物流情報管理モジュールは、第二の情報検証モジュールと、署名鍵情報生成モジュールとから構成され、第二の情報検証モジュールは、物流情報検証部と、第二の通信手段とから構成され、署名鍵情報生成モジュールは、署名鍵生成部と、第三の通信手段とから構成されるよ

うにしてもよい。

【００３７】

また、本発明の物流情報管理装置において、第一の検証鍵記憶手段と第二の検証鍵記憶手段に記憶される検証鍵は、全ての物流情報処理モジュールと物流情報管理モジュールとで共通としてもよい。

【００３８】

また、本発明の物流情報管理装置において、第一の情報検証手段と第二の情報検証手段とは、同じ動作をするようにしてもよい。

【００３９】

また、本発明の物流情報管理装置において、第一の情報検証部は、第一の情報検証手段が使用する検証鍵を選択する第一の検証鍵選択手段を持つようにしてもよい。（物流情報管理モジュールまたは物流管理センタごとに検証鍵を変えることができる。例えば、メーカ別のセンタがあって、仲介業者や小売店は異なるメーカの製品を扱うような場合が考えられる。公開鍵ＩＤで切り替える方法とセンタＩＤで切り替える方法が考えられる。）

また、本発明の物流情報管理装置において、第二の情報検証部は、第二の情報検証手段が使用する検証鍵を選択する第二の検証鍵選択手段を持つようにしてもよい。

【００４０】

また、本発明の物流情報管理装置において、署名鍵情報生成部は、署名鍵を選択する署名鍵選択手段を持つようにしてもよい。

【００４１】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されるようにしてもよい。

【００４２】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、検証鍵識別子とを含む情報を一単位として格納されるようにしてもよい。

【００４３】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、物流情報管理モジュール識別子と、を含む情報を一単位として格納されるようにしてもよい。

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つようにしてもよい。

【００４４】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つようにしてもよい。

【００４５】

また、本発明によれば、物品に添付され、上記物品に関する情報を記憶するデータキャリアに、上記物品の流通過程における１またはひとまとまりの取引ごとに生成される上記物品の流通情報（例えば取り扱い記録）と、１の上記流通情報の少なくとも一部または一連の上記流通情報のおのおのの少なくとも一部の署名値とを記憶するようにしている。

【００４６】

このようなデータキャリアを用いることにより偽物の混入等の不正を排除することができる。

【００４７】

上記物品の流通情報は、上記物品の識別子と、上記物品を受領した者の識別子と、上記署名値を生成する者の識別子とを少なくとも含むようにすることができる。

【００４８】

より具体的には、上記流通情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されるようにしてもよいし、少なくとも、検証鍵識別子とを含む情報を一単位として格納されるよう

にしてもよいし、少なくとも、物流情報管理モジュール識別子と、を含む情報を一単位として格納されるようにしてもよいし、少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つようにしてもよいし、さらに、少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つようにしてもよい。

【００４９】

また、本発明において、物流情報処理モジュールまたは物流情報管理モジュールの一方でのみ、署名の検証を行うようにしてもよい。

【００５０】

また、物流情報処理モジュールを情報読み取りおよび検証を行う部分と、情報を書き込み署名する部分とに分離しても良い。さらに物流情報管理モジュールを検証を行う部分と、署名鍵情報を管理し、物流情報処理モジュールに送る部分とに分離しても良い。

【００５１】

また、本発明は方法の形態で実現されても良く、また、少なくとも、一部が、コンピュータプログラム製品として実現されても良い。

【００５２】

【発明の実施の態様】

以下、本発明を実施例を用いて詳細に説明する。

【００５３】

［実施例１］

図１は、本実施例における構成を示している。図１において、本実施例の物流管理システムは、物品に添付され、物品に関する情報を記憶するデータキャリア１と、データキャリア１の情報を読み書きするための物流情報処理モジュール２と、物品の流通に関する情報を管理する物流情報管理モジュール３とから構成されている。

【００５４】

物流情報処理モジュール２は、データキャリア１のデータを読み出す読み取り手段４と、データキャリア１に情報を書込む書き込み手段５と、データキャリア１から読み出した情報を検証する第一の情報検証部６と、データキャリア１へ書込む情報を処理する情報生成部７と、物流情報管理モジュール３との通信を行うための第一の通信手段８とから構成されている。第一の情報検証部６は、データキャリア１から読み出した情報の検証を行う第一の情報検証手段９と、第一の情報検証手段９が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段１０とから構成されている。情報生成部７は、データキャリア１に書込む情報を生成する物流情報生成手段１１と、署名生成処理を行う署名モジュール１２と、署名生成に用いる署名鍵情報を選択するための署名鍵情報選択手段１３と、署名手段１６（署名モジュール１２）が電子署名生成時に使用する署名鍵情報を記憶する署名鍵情報記憶手段１４と、署名鍵情報を物流情報管理モジュール３から取得するための署名鍵情報取得手段１５とから構成されている。署名モジュール１２は、物流情報生成手段１１によって生成された情報に対する電子署名を生成する署名手段１６と、署名手段１６が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段１７とから構成されている。

【００５５】

物流情報管理モジュール３は、物流情報処理モジュール２との通信を行うための第二の通信手段１８と、物流情報処理モジュール２から送られてきた情報を処理するための第二の情報検証部１９と、物流情報処理モジュール２へ送る署名鍵情報を処理するための署名鍵情報生成部２０とから構成されている。第二の情報検証部１９は、物流情報処理モジュール２から送られてきた情報を検証するための第二の情報検証手段２１と、第二の情報検証手段２１が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段２２とから構成されている。署名鍵情報生成部２０は、物流情報処理モジュール２が物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段２３と、署名鍵情報生成手段２３が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段２４と、署名鍵情報生成手段２３が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段２５と、署名者秘密情報を記憶する第二の署名者秘密情報記憶手段２６とか

ら構成される。

【００５６】

図２は、本実施例においてデータキャリア１に記憶される物流情報の例を示している。データキャリア１には被署名部と署名値、必要ならその他の情報を一つのレコードとして物流情報が記憶されており、被署名部は少なくとも製品識別子、署名者識別子、受領者識別子を含んでいる。

【００５７】

図３は、データキャリア１から読み出した図２の物流情報を処理する場合の第一の情報検証部６での物流情報検証処理の流れを示している。

【００５８】

図３において、読み取り手段４によりデータキャリア１から読み出された物流情報が送られてくると、第一の情報検証手段９は、受領者識別子用の変数ｒｉｄをクリアする（ステップＳ３０１）。第一の署名検証鍵記憶手段１０から署名検証鍵を読み出し、変数Ｅ、変数ｎにセットする（ステップＳ３０２）。物流情報から最初のレコードを読み出す（ステップＳ３０３）。読み出したレコードから被署名部を読み出し、変数ｄａｔａにセットする（ステップＳ３０４）。ｄａｔａから製品識別子を読み出し、変数ｐｉｄにセットする（ステップＳ３０５）。ｄａｔａからから署名者識別子を読み出し、変数ｓｉｄにセットする（ステップＳ３０６）。関数Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）によってハッシュ値を計算し、変数ｈにセットする（ステップＳ３０７）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。ステップＳ３０３で読み出したレコードから署名部を読み出し、変数ｓｉｇｎにセットする（ステップＳ３０８）。検証用の値を次式によって計算し、変数ｖａｌにセットする（ステップＳ３０９）。

【００５９】

【数５】

ｓｉｇｎ^{E} ｍｏｄ　ｎ　（ｍｏｄ　は剰余演算）

ここで、ステップＳ３０９は署名アルゴリズムとしてＲＳＡ署名を用いている。ステップＳ３０９で他の署名アルゴリズムを用いる場合は、第一の署名検証鍵記憶手段１０に記憶され、ステップＳ３０８で読み込まれる鍵はそのアルゴリズムに適したものが使用される。ステップＳ３０７で計算したハッシュ値ｈとステップＳ３０９で計算した値ｖａｌが等しいか確認する。等しくなければ、署名がおかしい旨のエラーを返し、終了し、等しければ、ステップＳ３１１に進む（ステップＳ３１０）。ｒｉｄに値がセットされているか確認し（ステップＳ３１１）、ｒｉｄに値がセットされていなければ、ステップＳ３１３へ進み、ｒｉｄに値がセットされていれば、ステップＳ３０６で読み出した署名者識別子ｓｉｄと比較する（ステップＳ３１２）。ｒｉｄとｓｉｄが等しくなければ、レコードが不連続である旨のエラーを返し、終了し、等しければ、ステップＳ３１３に進む。処理すべきレコードが物流情報に残っているか、確認し（ステップＳ３１３）、残っていなければ検証に成功した旨を返し、終了し、残っていれば、ステップＳ３０４で読み出した被署名部ｄａｔａから受領者識別子を読み出し、変数ｒｉｄにセットし（ステップＳ３１４）、次のレコードを処理するためにステップＳ３０３に戻る。

【００６０】

本実施例ではステップＳ３０３からステップＳ３１４を繰り返すことにより、ステップＳ３０３で読み込む個々のレコード毎に署名の検証を行うが、個々のレコードの署名値ｓ［１］、ｓ［２］、…、ｓ［ｍ］（ｍはレコードの個数）、個々のレコードの被署名部に対するハッシュ値（ステップＳ３０７の計算結果）ｈ［１］、ｈ［２］、…、ｈ［ｍ］とステップＳ３０２で読み込む検証鍵Ｅ，ｎを使い次式が成り立つかどうかを確認することで、署名の検証を行うようにしてもよい。

【００６１】

【数６】

（ｓ［１］・ｓ［２］・…・ｓ［ｍ］）^{E}≡ｈ［１］・ｈ［２］・…・ｈ［ｍ］（ｍｏｄ　ｎ）

ステップＳ３１１およびステップＳ３１２は、ステップＳ３０６の直後で行うようにしても良い。

【００６２】

図４は、データキャリア１に記憶される物流情報に新しいレコードを追加する場合の物流情報処理モジュール２での処理の流れを示している。

【００６３】

図４において、まず、読み取り手段４によってデータキャリア１に物流情報が格納されているかどうかを確認し（ステップＳ４０１）、物流情報が格納されていなければ、物流情報生成手段１１は、新しいレコードの製品識別子ｐｉｄにデータキャリアを添付する製品の識別子をセットし（ステップＳ４０２ａ）、新しいレコードの署名者識別子ｓｉｄに署名者の識別子をセットし（ステップＳ４０３ａ）、ステップＳ４０７へ進む。ステップＳ４０２ａ、Ｓ４０３ａで与える識別子はユーザがその都度指定したり、適当なリストから読み込んだり、あらかじめ埋め込まれているものを用いるなど、既知の手法によって与えられるものとし、その方法は問わない。

【００６４】

ステップＳ４０１で物流情報が格納されていた場合、読み取り手段４によってデータキャリアから物流情報を読み込み（ステップＳ４０２ｂ）、読み込んだ物流情報から最後のレコードを読み出す（ステップＳ４０３ｂ）。読み出したレコードから被署名部を読み出し（ステップＳ４０４ｂ）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットする（ステップＳ４０５ｂ）。ステップＳ４０４ｂで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ４０６ｂ）。こののちステップＳ４０７へ進む。

【００６５】

さて、ステップ４０７において、新しいレコードの受領者識別子ｒｉｄに製品の出荷先（受領者）の識別子をセットする。新しいレコードの他のフィールドに適切な値をセットする（ステップＳ４０８）。ステップＳ４０７、Ｓ４０８でセットする値は、ステップＳ４０２ａ、Ｓ４０３ａ同様、ユーザが指定する、適当

なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数ｄａｔａにセットする（ステップＳ４０９）。署名手段１６は、署名者秘密情報記憶手段１７から署名者識別子を取出し、ｓｉｄ’にセットする（ステップＳ４０１０）。新しいレコードの署名者識別子ｓｉｄと、署名者秘密情報記憶手段１７から読み出した署名者識別子ｓｉｄ’とを比較し（ステップＳ４１１）、ステップＳ４１１でｓｉｄとｓｉｄ’が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、関数Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）によってハッシュ値を計算し、変数ｈにセットする（ステップＳ４１２）。ここで使われる関数Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）は、物流情報検証処理のハッシュ値の計算（図３のステップＳ３０７）で用いられるものと同じである。第一の署名者秘密情報記憶手段１７から署名者秘密情報を取出し変数ｄにセットする（ステップＳ４１３）。署名鍵情報選択手段１３は、署名鍵情報記憶手段１５から製品識別子ｐｉｄに対応する署名鍵情報を取出し、変数ｔ、ｎにセットする（ステップＳ４１４）。署名手段１６は、第一の署名値を次式によって計算し、変数ｒ１にセットする（ステップＳ４１５）。

【００６６】

【数７】

$$h^{f(d,n,pid,sid)} \mod n$$

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段１１は、第二の署名値を次式で計算する（ステップＳ４１６）。

【００６７】

【数８】

$$h^{t} \mod n$$

ステップＳ４１５、ステップＳ４１６の計算結果ｒ１とｒ２を使って、次式によ

り署名値を計算し、変数ｓｉｇｎにセットする（ステップＳ４１７）。

【００６８】

【数１０】

（ｒ１・ｒ２）　ｍｏｄ　ｎ

ステップＳ４１７で計算された署名値ｓｉｇｎを新しいレコードの署名値にセットする（ステップＳ４１８）。以上により物流情報に追加する新しいレコードが生成されたので、データキャリア１の物流情報に書き込み手段５により新しいレコードを追加する（ステップＳ４１９）。

【００６９】

本実施例では、署名鍵情報は既に署名鍵情報記憶手段１５に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップＳ４１３で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップＳ４１５は署名手段１６で行い、ステップＳ４１６は物流情報生成手段１１で行うが、ステップＳ４１６も署名手段１６で行うようにしても良いし、ステップＳ４１５とステップＳ４１６は署名手段１６と物流情報生成手段１１とで平行して計算するようにしてもよい。

【００７０】

図５は、署名鍵情報取得処置の流れを示している。署名鍵取得手段１５は、データキャリア１から物流情報を読み出す（ステップＳ５０１）。署名者秘密情報保持手段１７から署名者識別子を読み出す（ステップＳ５０２）。物流情報と署名者識別子を第一の通信手段８を通じて、物流情報管理モジュール３へ送る（ステップＳ５０３）。物流情報管理モジュール３では、第二の通信手段で受け取った物流情報を第二の物流情報検証手段１９に渡し、検証処理を行う（ステップＳ５０４）。検証処理は図３で示した第一の物流情報検証処理と同じである。ステップＳ５０４で検証に失敗した場合は、物流情報が正しくない旨を第二の通信手段１８を通じ、物流情報処理モジュール２へ送り、終了し、成功した場合は、署名者秘密情報選択手段２５は第二の署名者秘密情報記憶手段２６から第二の通信手段１８で受け取った署名者識別子ｓｉｄに対応する署名者秘密情報を読み出し

、変数ｄにセットする（ステップＳ５０５、Ｓ５０６）。署名鍵情報生成手段２３は署名鍵記憶手段から署名鍵を読み出し、変数Ｄ、ｎにセットする（ステップＳ５０７）。署名者識別子ｓｉｄとステップＳ５０４の検証処理中に読み出した製品識別子ｐｉｄおよびステップＳ５０７で読み出した署名鍵を使い次式を計算し、変数ｔへセットする（ステップＳ５０８）。

【００７１】

【数１１】

$$D-f(d, n, pid, sid)$$

ここでｆ（）は、前記署名生成処理のステップＳ４１５と同じ関数である。署名鍵情報としてｔ、ｎを第二の通信手段１８を通じて、物流情報処理モジュール３へ送る（ステップＳ５０９）。第一の通信手段１８を通じて、署名鍵情報を受け取ると、署名鍵情報取得手段１５は署名鍵情報記憶手段１４へ製品識別子と対応づけて格納する（ステップＳ５１０）。ここでＤは物流情報検証手段６で署名検証に使われるＥに対応する秘密情報であり、ｎはＥと共に検証に使われる公開情報である。署名アルゴリズムにＲＳＡ署名を用いる場合、

【００７２】

【数１２】

$$a^{E\cdot D} \mod n = a$$

なる関係を持つ値である。

【００７３】

また、署名鍵情報生成モジュールで、受け付けた製品識別子と署名者識別子の組を記録しておき、署名鍵情報生成処理を開始する前に、既に同じ製品識別子と署名者識別子の組を受け付けたことがないかを確認し、新しい組み合わせに対してのみ署名鍵情報を生成する、あるいは、再発行であることを記録に残すようにしてもよい。またこの記録をある製品識別子に付いて調べることで、その製品が流通過程でどこにあるかを知ることができ、特定の署名者識別子に付いて調べることで、その署名者の在庫状況を知ることが可能となる。

【００７４】

なお、物流情報のフォーマットとしては図２に示すものに限定されず、種々の

データ構造を採用できる。たとえば、図２に示すフォーマットにヘッダー情報を付加しても良い。すなわち、レコード１の前に検証鍵識別子を付与しても良い。図２のデータ構造を用いた処理においては検証鍵識別子は既知のものとして処理が行われているが、ヘッダー情報を付加することによりデータキャリア（タグ）ごとに検証鍵を付与することができる。この場合、図３のステップＳ３０２（検証鍵の読み出し）に代えて、ヘッダー情報に含まれる検証鍵識別子を用いて検証鍵を読み出すようにすればよい。

【００７５】

［実施例２］

図６は、第一の物流情報検証部６の他の構成例を示している。第一の物流情報検証部６は、第一の情報検証手段９と、第一の検証鍵記憶手段１０と、第一の検証鍵選択手段２９とから構成されている。

【００７６】

図７は、データキャリア１に記憶される物流情報の他の例を示している。データキャリア１には被署名部と署名値、必要ならその他の情報を一つのレコードとして物流情報が記憶されており、被署名部は少なくとも検証鍵識別子、製品識別子、署名者識別子、受領者識別子を含んでいる。

【００７７】

図８は、データキャリア１から読み出した図７の物流情報を処理する場合の本実施例における第一の情報検証部６での物流情報検証処理の流れを示している。

【００７８】

図８において、読み取り手段４によりデータキャリア１から読み出された物流情報が送られてくると、第一の情報検証手段９は、受領者識別子用の変数ｒｉｄをクリアする（ステップＳ８０１）。物流情報から最初のレコードを読み出す（ステップＳ８０２）。読み出したレコードから被署名部を読み出し、変数ｄａｔａにセットする（ステップＳ８０３）。ｄａｔａから製品識別子を読み出し、変数ｐｉｄにセットする（ステップＳ８０４）。ｄａｔａからから署名者識別子を読み出し、変数ｓｉｄにセットする（ステップＳ８０５）。関数Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）によってハッシュ値を計算し、変数ｈにセットする（ステップ

Ｓ８０６）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。ステップＳ８０２で読み出したレコードから署名部を読み出し、変数ｓｉｇｎにセットする（ステップＳ８０７）。ステップＳ８０３で読み出した被署名部ｄａｔａから検証鍵識別子を読み出し、変数ｋｉｄにセットする（ステップＳ８０８）。第一の署名検証鍵記憶手段１０から検証鍵識別子ｋｉｄに対応する署名検証鍵を読み出し、変数Ｅ、変数ｎにセットする（ステップＳ８０９）。検証用の値を次式によって計算し、変数ｖａｌにセットする（ステップＳ８１０）。

【００７９】

【数１３】

$sign^{E} \mod n$　（ｍｏｄは剰余演算）

ここで、ステップＳ８１０は署名アルゴリズムとしてＲＳＡ署名を用いている。ステップＳ８１０で他の署名アルゴリズムを用いる場合は、第一の署名検証鍵記憶手段１０に記憶され、ステップＳ８０９で読み込まれる鍵はそのアルゴリズムに適したものが使用される。ステップＳ８０６で計算したハッシュ値ｈとステップＳ８１０で計算した値ｖａｌが等しいか確認する。等しくなければ、署名がおかしい旨のエラーを返し、終了し、等しければ、ステップＳ８１２に進む（ステップＳ８１１）。ｒｉｄに値がセットされているか確認し（ステップＳ８１２）、ｒｉｄに値がセットされていなければ、ステップＳ８１４へ進み、ｒｉｄに値がセットされていれば、ステップＳ８０５で読み出した署名者識別子ｓｉｄと比較する。ｒｉｄとｓｉｄが等しくなければ、レコードが不連続である旨のエラーを返し、終了し、等しければ、ステップＳ８１４に進む（ステップＳ８１３）。処理すべきレコードが物流情報に残っているか、確認し（ステップＳ８１４）、残っていなければ検証に成功した旨を返し、終了し、残っていれば、ステップＳ８０３で読み出した被署名部ｄａｔａから受領者識別子を読み出し、変数ｒｉｄにセットし（ステップＳ８１５）、次のレコードを処理するためにステップＳ８

０２に戻る。

【００８０】

ステップＳ８１２およびステップＳ８１３はステップＳ８０６の前に行うようにしても良い。

【００８１】

［実施例３］

図９は、署名モジュール１２の他の構成例を示している。署名モジュール１２は、署名手段１６と第一の署名者秘密情報記憶手段１７と署名鍵使用限度保持手段２７とから構成されている。

【００８２】

図１０は、本実施例における署名生成処理の流れを示している。図１０において、読み取り手段４によってデータキャリア１に物流情報が格納されているかどうかを確認し（ステップＳ１００１）、物流情報が格納されていなければ、物流情報生成手段１１は、新しいレコードの製品識別子ｐｉｄにデータキャリアを添付する製品の識別子をセットし（ステップＳ１００２ａ）、新しいレコードの署名者識別子ｓｉｄに署名者の識別子をセットし（ステップＳ１００３ａ）、ステップＳ１００７へ進む。ステップＳ１００２ａ、Ｓ１００３ａで与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップＳ１００１で物流情報が格納されていた場合、読み取り手段４によってデータキャリアから物流情報を読み込み（ステップＳ１００２ｂ）、読み込んだ物流情報から最後のレコードを読み出す（ステップＳ１００３ｂ）。読み出したレコードから被署名部を読み出し（ステップＳ１００４ｂ）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットする（ステップＳ１００５ｂ）。ステップＳ１００４ｂで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ１００６ｂ）。

【００８３】

ステップＳ１００３ａまたはステップＳ１００６ｂに続いて、新しいレコード

の受領者識別子ｒｉｄに製品の出荷先（受領者）の識別子をセットする（ステップＳ１００７）。新しいレコードの他のフィールドに適切な値をセットする（ステップＳ１００８）。ステップＳ１００７、Ｓ１００８でセットする値は、ステップＳ１００２ａ、Ｓ１００３ａ同様、ユーザが指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数ｄａｔａにセットする（ステップＳ１００９）。署名手段１６は、署名者秘密情報記憶手段１７から署名者識別子を取出し、ｓｉｄ’にセットする（ステップＳ１０１０）。新しいレコードの署名者識別子ｓｉｄと、署名者秘密情報記憶手段１７から読み出した署名者識別子ｓｉｄ’とを比較し（ステップＳ１０１１）、ステップＳ１０１１でｓｉｄとｓｉｄ’が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、署名鍵使用限度情報保持手段２７から製品識別子ｐｉｄに対応する署名鍵情報の署名鍵使用限度情報を読み出し、変数ｍにセットする（ステップＳ１０１２）。ｍが正の値でなければ、該当する署名鍵は規定の使用回数に達している旨のエラーを返し、終了し、ｍが正の値であれば、ステップＳ１０１４に進み（ステップＳ１０１３）、ｍから１減じた値をｍにセットする（ステップＳ１０１４）。署名鍵使用限度情報保持手段２７に記憶されている署名鍵使用限度情報を変数ｍの値に更新する（ステップＳ１０１５）。ステップＳ１００２ａまたはステップＳ１００５ｂでセットした製品識別子ｐｉｄ、ステップＳ１００３ａまたはステップＳ１００６ｂでセットした署名者識別子ｓｉｄ、ステップＳ１００９でセットした被署名部ｄａｔａをもとにハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ１０１６）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。署名手段１６は第一の署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットする（ステップＳ１０１７）。署名鍵情報選択手段１３は製品識別子ｐｉｄに対応する署名鍵情報を署名鍵

情報記憶手段１４から読み出し、変数ｔ、ｎにセットする（ステップＳ１０１８）。署名手段１６は、第一の署名値を次式により計算し、変数ｒ１にセットする（ステップＳ１０１９）。

【００８４】

【数１４】

$$h^{f(d,n,pid,sid)} \quad mod \quad n$$

ここで、関数ｆ（）は引数ｄ、　ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段１１は、第二の署名値を次式で計算し、変数ｒ２にセットする（ステップＳ１０２０）。

【００８５】

【数１５】

$$h^{t} \quad mod \quad n$$

ステップＳ１０１９、ステップＳ１０２０の計算結果ｒ１、ｒ２を使って次式により署名値を計算し、変数ｓｉｇｎにセットする（ステップＳ１０２１）。

【００８６】

【数１６】

$$(r1 \cdot r2) \quad mod \quad n$$

ステップＳ１０２１で計算された署名値ｓｉｇｎを新しいレコードの署名値にセットし（ステップＳ１０２２）、上のステップで生成された新しいレコードを書き込み手段５によりデータキャリア１内の物流情報に追加する（ステップＳ１０２３）。

【００８７】

本実施例では署名鍵情報は既に署名鍵情報記憶手段１５に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップＳ１０１８で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップＳ１０１９は署

名手段１６で、ステップＳ１０２０は物流情報生成手段１１で行うが、ステップＳ１０２０も署名手段１６で行うようにしても良いし、ステップＳ１０１９とステップＳ１０２０は署名手段１６と物流情報生成手段１１とで平行して計算するようにしてもよい。

【００８８】

［実施例４］

図１１は、図９の署名モジュール１２の処理の流れを示す他の実施例である。図１１において、読み取り手段４によってデータキャリア１に物流情報が格納されているかどうかを確認し（ステップＳ１１０１）、物流情報が格納されていなければ、物流情報生成手段１１は、新しいレコードの製品識別子ｐｉｄにデータキャリアを添付する製品の識別子をセットし（ステップＳ１１０２ａ）、新しいレコードの署名者識別子ｓｉｄに署名者の識別子をセットし（ステップＳ１１０３ａ）、ステップＳ１１０７へ進む。ステップＳ１１０２ａ、Ｓ１１０３ａで与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップＳ１１０１で物流情報が格納されていた場合、読み取り手段４によってデータキャリアから物流情報を読み込み（ステップＳ１１０２ｂ）、読み込んだ物流情報から最後のレコードを読み出す（ステップＳ１１０３ｂ）。読み出したレコードから被署名部を読み出し（ステップＳ１１０４ｂ）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットする（ステップＳ１１０５ｂ）。ステップＳ１１０４ｂで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ１１０６ｂ）。新しいレコードの受領者識別子ｒｉｄに製品の出荷先（受領者）の識別子をセットする（ステップＳ１１０７）。新しいレコードの他のフィールドに適切な値をセットする（ステップＳ１１０８）。ステップＳ１１０７、Ｓ１１０８でセットする値は、ステップＳ１１０２ａ、Ｓ１１０３ａ同様、ユーザが指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数ｄａｔａにセットする

（ステップＳ１１０９）。署名手段１６は、署名者秘密情報記憶手段１７から署名者識別子を取出し、ｓｉｄ’にセットする（ステップＳ１１１０）。新しいレコードの署名者識別子ｓｉｄと、署名者秘密情報記憶手段１７から読み出した署名者識別子ｓｉｄ’とを比較し（ステップＳ１１１１）、ステップＳ１１１１でｓｉｄとｓｉｄ’が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、署名鍵使用限度情報保持手段２７から製品識別子ｐｉｄに対応する署名鍵情報の署名鍵使用限度情報の有無を確認する（ステップＳ１１１２）。署名鍵使用限度情報があれば、該当する署名鍵は使用済である旨のエラーを返し、終了し、署名鍵使用限度情報がなければ、ステップＳ１１１３に進み、製品識別子に対応されて署名鍵使用限度情報を署名鍵使用限度情報保持手段２７に登録する（ステップＳ１１１３）。ステップＳ１１０２ａまたはステップＳ１１０５ｂでセットした製品識別子ｐｉｄ、ステップＳ１１０３ａまたはステップＳ１１０６ｂでセットした署名者識別子ｓｉｄ、ステップＳ１１０９でセットした被署名部ｄａｔａをもとにハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ１１１４）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。署名手段１６は第一の署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットする（ステップＳ１１１５）。署名鍵情報選択手段１３は製品識別子ｐｉｄに対応する署名鍵情報を署名鍵情報記憶手段１４から読み出し、変数ｔ、ｎにセットする（ステップＳ１１１６）。署名手段１６は、第一の署名値を次式により計算し、変数ｒ１にセットする（ステップＳ１１１７）。

【００８９】

【数１７】

$$h^{f(d,n,pid,sid)} \mod n$$

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数であ

る。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段１１は、第二の署名値を次式で計算し、変数ｒ２にセットする（ステップＳ１１１８）。

【００９０】

【数１８】

$h^t \mod n$

ステップＳ１１１７、ステップＳ１１１８の計算結果ｒ１、ｒ２を使って次式により署名値を計算し、変数ｓｉｇｎにセットする（ステップＳ１１１９）。

【００９１】

【数１９】

$(r1 \cdot r2) \mod n$

ステップＳ１１１９で計算された署名値ｓｉｇｎを新しいレコードの署名値にセットし（ステップＳ１１２０）、上のステップで生成された新しいレコードを書き込み手段５によりデータキャリア１内の物流情報に追加する（ステップＳ１１２１）。本実施例では署名鍵情報は既に署名鍵情報記憶手段１５に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップＳ１１１６で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップＳ１１１７は署名手段１６で、ステップＳ１１１８は物流情報生成手段１１で行うが、ステップＳ１１１８も署名手段１６で行うようにしても良いし、ステップＳ１１１７とステップＳ１１１８は署名手段１６と物流情報生成手段１１とで平行して計算するようにしてもよい。

【００９２】

［実施例５］

図１２は、署名鍵情報生成部２０の他の構成例を示している。署名鍵情報生成部２０は、署名鍵情報生成手段２３と署名者秘密情報選択手段２５と第二の署名者秘密鍵情報記憶手段２６と署名鍵記憶手段２４と署名鍵選択手段３１とから構成されている。

【００９３】

図１３は、署名鍵情報取得処理の流れを示している。図１３において、署名鍵取得手段１５は、データキャリア１から物流情報を読み出す（ステップＳ１３０１）。署名者秘密情報保持手段１７から署名者識別子を読み出す（ステップＳ１３０２）。物流情報と署名者識別子を第一の通信手段８を通じて、物流情報管理モジュール３へ送る（ステップＳ１３０３）。物流情報管理モジュール３では、第二の通信手段１８で受け取った物流情報を第二の物流情報検証手段１９に渡し、検証処理を行う（ステップＳ１３０４）。検証処理は図８で示した第一の物流情報検証処理と同じである。ステップＳ１３０４で検証に失敗した場合は、物流情報が正しくない旨を第二の通信手段１８を通じ、物流情報処理モジュール２へ送り、終了し、成功した場合は、署名者秘密情報選択手段２５は第二の署名者秘密情報記憶手段２６から第二の通信手段１８で受け取った署名者識別子に対応する署名者秘密情報を読み出し、変数ｄにセットする（ステップＳ１３０５、Ｓ１３０６）。署名鍵情報生成手段２３は、ステップＳ１３０４の検証処理中に取りだした検証鍵識別子ｋｉｄに対応する署名鍵を署名鍵記憶手段から読み出し、変数Ｄ、ｎにセットする（ステップＳ１３０７）。署名者識別子ｓｉｄとステップＳ１３０４の検証処理中に読み出した製品識別子ｐｉｄおよびステップＳ１３０７で読み出した署名鍵を使い次式を計算し、変数ｔへセットする（ステップＳ１３０８）。

【００９４】

【数２０】

$$D-f(d, n, pid, sid)$$

ここでｆ（）は、前記署名生成処理（図１０のステップＳ１０１９や図１１のステップＳ１１１７）と同じ関数である。署名鍵情報としてｔ、ｎ、ｋｉｄを第二の通信手段１８を通じて、物流情報処理モジュール３へ送る（ステップＳ１３０９）。第一の通信手段１８を通じて、署名鍵情報を受け取ると、署名鍵情報取得手段１５は署名鍵情報記憶手段１４へ製品識別子と対応づけて格納する（ステップＳ１３１０）。ここでＤは物流情報検証手段６で署名検証に使われるＥに対応する秘密情報であり、ｎはＥと共に検証に使われる公開情報である。署名アルゴ

リズムにＲＳＡ署名を用いる場合、

【００９５】

【数２１】

$$a^{E \cdot D} \mod n = a$$

なる関係を持つ値となる。

【００９６】

［実施例６］

図１４は、データキャリア１に格納される物流情報の一実施例を示している。データキャリア１には被署名部と必要ならその他の情報を一つのレコードとして、複数のレコードからなるデータ部と、署名値からなる物流情報が記憶されており、被署名部は少なくとも製品識別子、署名者識別子、受領者識別子を含んでいる。

【００９７】

図１５は、図１４に示される物流情報を検証するために物流情報検証部６で行われる検証処理の流れを示している。図１５において、データキャリア１から読み取られた物流情報が与えられると、変数ｖａｌを初期化し（１をセット。ステップＳ１５０１）、第一の検証鍵記憶手段１０から署名検証鍵を読み出し、変数Ｅ，ｎにセットする（ステップＳ１５０２）。物流情報からデータ部を読み出す（ステップＳ１５０３）。読み出したデータ部から１レコード読み出す（ステップＳ１５０４）、読み出したレコードから被署名部を取出し、変数ｄａｔａにセットする（ステップＳ１５０５）。被署名部から製品識別子を読み出し、変数ｐｉｄにセットし（ステップＳ１５０６）、被署名部から署名者識別子を取出し、変数ｓｉｄにセットする（ステップＳ１５０８）。ステップＳ１５０５で読み出した被署名部ｄａｔａ、ステップＳ１５０６で読み出した製品識別子ｐｉｄ、ステップＳ１５０７で読み出した署名者識別子ｓｉｄを使いハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ１５０８）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を

適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。次式を計算し、変数ｖａｌにセットする（ステップＳ１５０９）。

【００９８】

【数２２】

$$(val \cdot h) \mod n$$

ステップＳ１５０３で読み出したデータ部に未処理のレコードがあるか確認する（ステップＳ１５１０）。未処理のレコードがあればステップＳ１５０４へ戻り、次のレコードを処理する。未処理のレコードがなければ、物流情報から署名値を読み出し、変数ｓｉｇｎにセットする（ステップＳ１５１１）。ステップＳ１５０２で読み出した検証鍵Ｅ，ｎとステップＳ１５１１で読み出した署名値ｓｉｇｎを使って、次式により検証値を計算し、変数ｖａｌ２にセットする（ステップＳ１５１２）。

【００９９】

【数２３】

$$sign^{E} \mod n$$

変数ｖａｌとｖａｌ２の値を比較し（ステップＳ１５１３）、等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【０１００】

図１６は、図１４に示される物流情報を生成するために物流情報生成部６における処理の流れを示している。データキャリア１に物流情報があるか確認し（ステップＳ１６０１）、物流情報が格納されていなければ、物流情報生成手段１１は、新しいレコードの製品識別子ｐｉｄにデータキャリアを添付する製品の識別子をセットし（ステップＳ１６０２ａ）、新しいレコードの署名者識別子ｓｉｄに署名者の識別子をセットし（ステップＳ１６０３ａ）、ステップＳ１６０８に進む。ステップＳ１６０２ａ、Ｓ１６０３ａで与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップＳ１６０１で物流情報が格納されていた場合、読み取り手段４によってデータキャリアから物流情

報を読み込み（ステップＳ１６０２ｂ）、読み込んだ物流情報からデータ部を読み出す（ステップＳ１６０３ｂ）。読み込んだデータ部から最後のレコードを読み出し（ステップＳ１６０４ｂ）、読み出したレコードから被署名部を読み出す（ステップＳ１６０５ｂ）。その被署名部から製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットする（ステップＳ１６０６ｂ）。ステップＳ１６０４ｂで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ１６０７ｂ）。新しいレコードの受領者識別子ｒｉｄに製品の出荷先（受領者）の識別子をセットする（ステップＳ１６０８）。新しいレコードの他のフィールドに適切な値をセットする（ステップＳ１６０９）。新しいレコードから被署名部を取出し、変数ｄａｔａへセットする（ステップＳ１６１０）。署名手段１６は、署名者秘密情報記憶手段１７から署名者識別子を取出し、ｓｉｄ’にセットする（ステップＳ１６１１）。新しいレコードの署名者識別子ｓｉｄと、署名者秘密情報記憶手段１７から読み出した署名者識別子ｓｉｄ’とを比較し（ステップＳ１６１２）、ステップＳ１６１２でｓｉｄとｓｉｄ’が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、ステップＳ１６０２ａまたはステップＳ１６０６ｂでセットした製品識別子ｐｉｄ、ステップＳ１６０３ａまたはステップＳ１６０７ｂでセットした署名者識別子ｓｉｄ、ステップＳ１６１０でセットした被署名部ｄａｔａをもとにハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ１６１３）。ここでハッシュ関数Ｈ（）は、物流情報検証部６で行われる検証処理（図１５のステップＳ１５０８）で使われるハッシュ関数と同じである。署名手段１６は第一の署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットする（ステップＳ１６１４）。署名鍵情報選択手段１３は製品識別子ｐｉｄに対応する署名鍵情報を署名鍵情報記憶手段１４から読み出し、変数ｔ、ｎにセットする（ステップＳ１６１５）。署名手段１６は、第一の署名値を次式により計算し、変数ｒ１にセットする（ステップＳ１６１６）。

【０１０１】

【数２４】

$h^{f(d,n,pid,sid)}$ ｍｏｄ ｎ

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段１１は、第二の署名値を次式で計算し、変数ｒ２にセットする（ステップＳ１６１７）。

【０１０２】

【数２５】

h^{t} ｍｏｄ ｎ

ステップＳ１６１６、ステップＳ１６１７の計算結果ｒ１、ｒ２を使って次式により署名値を計算し、変数ｓにセットする（ステップＳ１６１８）。

【０１０３】

【数２６】

（ｒ１・ｒ２） ｍｏｄ ｎ

ステップＳ１６０２ｂで読み出した物流情報から署名部を読み出し、変数ｓｉｇｎにセットする（ステップＳ１６１９）。ステップＳ１６１５で読み出したｎ、ステップＳ１６１８で計算したｓ、ステップＳ１６１９で読み出したｓｉｇｎを使い、次式を計算し、変数ｓｉｇｎにセットする（ステップＳ１６２０）。

【０１０４】

【数２７】

（ｓｉｇｎ・ｓ） ｍｏｄ ｎ

データキャリア１のデータ部に上記ステップで生成された新しいレコードを追加し（ステップＳ１６２１）、ステップＳ１６２０で計算された変数ｓｉｇｎの値でデータキャリア１の署名値を更新する（ステップＳ１６２２）。

【０１０５】

本実施例では署名鍵情報は既に署名鍵情報記憶手段１５に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップＳ１６１５で署名鍵情報が見つからない場合に、署名情報取得処理を実行す

ることにより、取得しても良い。また、上記手順では、ステップＳ１６１６は署名手段１６で、ステップＳ１６１７は物流情報生成手段１１で行うが、ステップＳ１６１７も署名手段１６で行うようにしても良いし、ステップＳ１６１６とステップＳ１６１７は署名手段１６と物流情報生成手段１１とで平行して計算するようにしてもよい。

【０１０６】

［実施例７］

図１７は、図１４に示される物流情報を検証するために物流情報検証部６で行われる検証処理の流れを示している。図１７において、読み取り手段４によりデータキャリア１から物流情報を読み取り（ステップＳ１７０１）、読み取った物流情報からデータ部を読み出し（ステップＳ１７０２）、変数ｒｉｄ，ｄａｔａをクリアする（ステップＳ１７０３）。ステップＳ１７０２で読み出したデータ部から１レコード読み出し（ステップＳ１７０４）、読み出したレコードから被署名部を読み出し、変数ｄにセットする（ステップＳ１７０５）。ステップＳ１７０５で読み出した被署名部から署名者識別子を読み出し、変数ｓｉｄにセットする（ステップＳ１７０６）。変数ｒｉｄに値がセットされているか確認し（ステップＳ１７０７）、セットされていなければ、ステップＳ１７０９に進み、セットされていれば、ステップＳ１７０６で読み出した署名者識別子ｓｉｄとｒｉｄを比較する（ステップＳ１７０８）。ｓｉｄとｒｉｄが等しくなければ、レコードが不連続である旨を返し、終了し、等しければステップＳ１７０９に進む。ステップＳ１７０５で読み出した被署名部ｄと変数ｄａｔａの値を使って次式を計算し、変数ｄａｔａにセットする（ステップＳ１７０９）。

【０１０７】

【数２８】

ｄａｔａ｜ｄ　（｜はビット列の連結。ただしｄａｔａがクリアされている時は、ｄを返す）

ここでステップＳ１７０９での計算は他の方法でもよく、ｄａｔａとｄを使った計算結果がｄａｔａとｄの組に対して一対一に対応していればよい。ステップＳ１７０２で読み出したデータ部に未処理のレコードがあるか確認し（ステップＳ

１７１０）、未処理のレコードがあれば、ステップＳ１７０５で読み出した被署名部から受領者識別子を変数ｒｉｄにセットし（ステップＳ１７１１）、ステップＳ１７０４に戻り、次の未処理レコードの処理をおこない、未処理のレコードが無ければ、ステップＳ１７１２に進む。ステップＳ１７０５で読み出した被署名部ｄから製品識別子を読み出し、変数ｐｉｄにセットする（ステップＳ１７１２）。上記ステップにより計算された変数ｄａｔａ、ステップＳ１７０６で読み出した署名者識別子ｓｉｄ、ステップＳ１７１２で読み出した製品識別子ｐｉｄを使いハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ１７１３）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。ステップＳ１７０１で読み出した物流情報から署名値を読み出し、変数ｓｉｇｎにセットする（ステップＳ１７１４）。第一の検証鍵記憶手段１０から署名検証鍵を読み出し、変数Ｅ，ｎにセットする（ステップＳ１７１５）。ステップＳ１７１４で読み出した署名値ｓｉｇｎとステップＳ１７１５で読み出した署名検証鍵Ｅ，ｎを使い次式により検証用の値を計算し、変数ｖａｌにセットする（ステップＳ１７１６）。

【０１０８】

【数２９】

$$sign^{E} \mod n$$

ステップＳ１７１３で計算したハッシュ値ｈとステップＳ１７１６で計算した検証用の値ｖａｌとを比較する（ステップＳ１７１７）。ｈとｖａｌが等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【０１０９】

図１８は、図１４に示される物流情報を生成するために物流情報生成部７で行われる処理の流れを示している。図１８において、変数ｄａｔａを初期化し（ステップＳ１８０１）、データキャリア１内に物流情報があるか確認し（ステップ

Ｓ１８０２）、無ければ、物流情報のデータ部に記載するための新しいレコードの製品識別子ｐｉｄをセットし（ステップＳ１８０３ａ）、新しいレコードの署名者識別子ｓｉｄをセットし（ステップ１８０４ａ）、ステップＳ１８１３へ進む。ステップＳ１８０２で物流情報があれば、データキャリア１から読み取り手段４を介し物流情報を読み出し（ステップＳ１８０３ｂ）、読み出した物流情報からデータ部を取出し（ステップＳ１８０４ｂ）、そのデータ部から１レコード取出す（ステップＳ１８０５ｂ）。取出したレコードの被署名部を取出し、変数ｄにセットする（ステップＳ１８０６ｂ）。変数ｄａｔａとステップＳ１８０６ｂで取出した被署名部ｄとから次式の計算を行い、変数ｄａｔａにセットする（ステップＳ１８０７ｂ）。

【０１１０】

【数３０】

ｄａｔａ｜ｄ　（｜はビット列の連結。ただしｄａｔａがクリアされている時は、ｄを返す）

ステップＳ１８０７ｂの計算は、他の方式でも良いが、物流情報検証部６で行われる検証処理での計算（図１７のステップＳ１７０９）と同じであればよい。ステップＳ１８０４ｂで読み出したデータ部に未処理のレコードがあるか確認し（ステップＳ１８０８ｂ）、未処理のレコードが無ければステップＳ１８０９ｂに進み、未処理のレコードがあれば、ステップＳ１８０５ｂに戻り、次のレコードを処理する。上記ステップにより読み出された被署名部ｄから製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットし（ステップＳ１８０９ｂ）、被署名部ｂから受領者識別子ｒｉｄを読み出す（ステップＳ１８１０ｂ）。署名者秘密情報保持手段１７から署名者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ１８１１ｂ）。ステップ１０ｂで読み出した受領者識別子ｒｉｄとステップＳ１８１１ｂで読み出した署名者識別子ｓｉｄを比較し（ステップＳ１８１２ｂ）、受領者識別子ｒｉｄと署名者識別子ｓｉｄが等しければ、ステップＳ１８１３へ進み、等しくなければ受領者と署名者が一致しない旨を返し、終了する。製品の受領者の識別子を新しいレコードの受領者識別子ｒｉｄにセットする（ステップＳ１８１３）。新しいレコードの残り

のフィールドをセットし（ステップＳ１８１４）、新しいレコードの被署名部を取り出し、変数ｄへセットする（ステップＳ１８１５）。変数ｄａｔａと変数ｄに対し、ステップＳ１８０７ｂと同じ計算を行い、変数ｄａｔａにセットする（ステップＳ１８１６）。ステップＳ１８０３ａまたはステップＳ１８０９ｂでセットした製品識別子ｐｉｄと、ステップＳ１８０４ａまたはステップＳ１８１１ｂでセットした署名者識別子ｓｉｄと、ステップＳ１８１６でセットした変数ｄａｔａに対し、ハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算する（ステップＳ１８１７）。ここでハッシュ関数Ｈ（）は、物流情報検証部６で行われる検証処理（図１７のステップＳ１７１３）で用いられるものと同じである。署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットし（ステップＳ１８１８）、署名鍵情報記憶手段１４から製品識別子ｐｉｄに対応する署名鍵情報を取出し、変数ｔ，ｎにセットする（ステップＳ１８１９）。次式により、第一の署名値を計算し、変数ｓ１にセットする（ステップＳ１８２０）。

【０１１１】

【数３１】

$$h^{f(d,n,pid,sid)} \mod n$$

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。次式により、第二の署名値を計算し、変数ｓ２にセットする（ステップＳ１８２１）。

【０１１２】

【数３２】

$$h^{t} \mod n$$

ステップＳ１８２０で計算された第一の署名値ｓ１、ステップＳ１８２１で計算された第二の署名値ｓ２を使い次式により署名値を計算し、変数ｓｉｇｎにセットする（ステップＳ１８２２）。

【０１１３】

【数３３】

ｓ１・ｓ２　ｍｏｄ　ｎ

書き込み手段５を介し、データキャリア１内のデータ部に新しいレコードを追加し（ステップＳ１８２３）、データキャリア１内の署名値をステップＳ１８２２で計算した変数ｓｉｇｎの値で更新する（ステップＳ１８２４）。

【０１１４】

［実施例８］

図１９は、データキャリア１に格納される物流情報の一実施例を示している。データキャリア１には被署名部と必要ならその他の情報を一つのレコードとして、複数のレコードからなるデータ部と、検証鍵識別子と署名値を一つのレコードとして、複数のレコードからなる署名値部とからなる物流情報が記憶されており、被署名部は少なくとも検証鍵識別子、製品識別子、署名者識別子、受領者識別子を含んでいる。

【０１１５】

図２０は、図１９に示される物流情報を検証するために物流情報検証部６で行われる検証処理の流れを示している。図２０において、データキャリアからデータ部を読み出す（ステップＳ２００１）、読み出したデータ部から１レコード読み出す（ステップＳ２００２）。読み出したレコードから、被署名部を取り出し、変数ｄａｔａにセットする（ステップＳ２００３）。被署名部から、製品識別子を読み出し、変数ｐｉｄにセットする（ステップＳ２００４）。被署名部から、受領者識別子を読み出し、変数ｓｉｄにセットする（ステップＳ２００５）。ステップＳ２００３で読み出した被署名部ｄａｔａ、ステップＳ２００４で読み出した製品識別子ｐｉｄ、ステップＳ２００５で読み出した署名者識別子ｓｉｄを使いハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ２００６）。被署名部から検証鍵識別子を読み出し、変数ｋｉｄにセットする（ステップＳ２００７）。検証鍵識別子ｋｉｄに対応するバッファ変数ｖａｌ［ｋｉｄ］が存在するかを確認し、存在しない場合は領域を確保する（ステップＳ２００８、Ｓ２００９）。変数ｖａｌ［ｋｉｄ］に１をセットする（ステップＳ２０１０）。署名検証鍵記憶手段２４からｋｉｄに対応する署名検

証鍵法数を読み出し、変数ｎにセットする（ステップＳ２０１１）。次式を計算し、変数ｖａｌ［ｋｉｄ］にセットする（ステップＳ２０１２）。

【０１１６】

【数３４】

（ｖａｌ［ｋｉｄ］・ｈ）　ｍｏｄ　ｎ

ステップＳ２００１で読み出したデータ部に未処理のレコードがあるか確認する（ステップＳ２０１３）。未処理のレコードがあればステップＳ２００２へ戻り、次のレコードを処理する。未処理のレコードがなければ、物流情報から署名値部を読み出す（ステップＳ２０１４）。読み出した署名値部から１レコード読み出す（ステップ１５）。被署名部から検証鍵識別子を読み出し、変数ｋｉｄにセットする（ステップＳ２０１６）。署名検証鍵記憶手段２４から検証鍵識別子ｋｉｄに対応する署名検証鍵を読み出し、変数Ｅ、ｎにセットする（ステップＳ２０１７）。ステップＳ２０１５で読み出したレコードから署名値を読み出し、変数ｓｉｇｎにセットする（ステップＳ２０１８）。ステップＳ２０１７で読み出した検証鍵Ｅ，ｎとステップＳ２０１８で読み出した署名値ｓｉｇｎを使って、次式により検証値を計算し、変数ｖａｌ２にセットする（ステップＳ２０１９）。

【０１１７】

【数３５】

$sign^{E}$　ｍｏｄ　ｎ

変数ｖａｌ［ｋｉｄ］とｖａｌ２の値を比較し（ステップＳ２０２０）、等しくなければ検証に失敗した旨を返し、終了する。等しければ、ステップＳ２０１４で読み出した署名値部に未処理のレコードがあるか確認する（ステップＳ２０２１）。未処理のレコードがあればステップＳ２０１５へ戻り、次のレコードを処理する。未処理のレコードがなければ、検証に成功した旨を返し、終了する。

【０１１８】

図２１は、図１９に示される物流情報を生成するために物流情報生成部６における処理の流れを示している。図２１において、データキャリア１に物流情報があるか確認し（ステップＳ２１０１）、物流情報が格納されていなければ、物流

情報生成手段１１は、新しいレコードの製品識別子ｐｉｄにデータキャリアを添付する製品の識別子をセットし（ステップＳ２１０２ａ）、新しいレコードの署名者識別子ｓｉｄに署名者の識別子をセットし（ステップＳ２１０３ａ）、ステップＳ２１１０に進む。ステップＳ２１０２ａ、Ｓ２１０３ａで与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップＳ２１０１で物流情報が格納されていた場合、読み取り手段４によってデータキャリアから物流情報を読み込み（ステップＳ２１０２ｂ）、読み込んだ物流情報からデータ部を読み出す（ステップＳ２１０３ｂ）。読み込んだデータ部から最後のレコードを読み出し（ステップＳ２１０４ｂ）、読み出したレコードから被署名部を読み出す（ステップＳ２１０５ｂ）。その被署名部から製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットする（ステップＳ２１０６ｂ）。ステップＳ２１０４ｂで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ２１０７ｂ）。署名手段１６は、署名者秘密情報記憶手段１７から署名者識別子を取出し、ｓｉｄ’にセットする（ステップＳ２１０８ｂ）。新しいレコードの署名者識別子ｓｉｄと、署名者秘密情報記憶手段１７から読み出した署名者識別子ｓｉｄ’とを比較し（ステップＳ２１０９ｂ）、ステップＳ２１０９ｂでｓｉｄとｓｉｄ’が等しくなければ、署名者がおかしい旨のエラーを返し、終了する。新しいレコードの受領者識別子ｒｉｄに製品の出荷先（受領者）の識別子をセットする（ステップＳ２１１０）。署名鍵記憶手段１４から署名鍵情報を読み出し、署名鍵情報をｔへ、署名鍵法数をｎへ、署名鍵識別子をｉｄへセットする（ステップＳ２１１１）。新しいレコードの検証鍵識別子ｋｉｄに署名鍵識別子ｉｄをセットする（ステップＳ２１１２）。新しいレコードの他のフィールドに適切な値をセットする（ステップＳ２１１３）。新しいレコードから被署名部を取出し、変数ｄａｔａへセットする（ステップＳ２１１４）。ステップＳ２１０２ａまたはステップＳ２１０６ｂでセットした製品識別子ｐｉｄ、ステップＳ２１０３ａまたはステップＳ２１０７ｂでセットした署名者識別子ｓｉｄ、ステップＳ２１１４でセットした被署名部ｄａｔａをもとにハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ

）を計算し、変数ｈにセットする（ステップＳ２１１５）。署名手段16は第一の署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットする（ステップＳ２１１６）。署名手段１６は、第一の署名値を次式により計算し、変数ｒ１にセットする（ステップＳ２１１７）。

【０１１９】

【数３６】

$$h^{f(d,n,pid,sid)} \mod n$$

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段１１は、第二の署名値ｒ２を次式で計算する（ステップＳ２１１８）。

【０１２０】

【数３７】

$$h^{t} \mod n$$

ステップＳ２１１７、ステップＳ２１１８の計算結果ｒ１、ｒ２を使って次式により署名値を計算し、変数ｓにセットする（ステップＳ２１１９）。

【０１２１】

【数３８】

$$(r1 \cdot r2) \mod n$$

データキャリア１から署名値部を読み出す（ステップＳ２１２０）。検証鍵識別子ｋｉｄに対応する署名値があるか確認し、ある場合は、データキャリア１から署名値を読み出し、変数ｓｉｇｎにセットし（ステップＳ２１２１、Ｓ２１２２ａ）、ステップＳ２１１９でセットしたｓと署名鍵法数ｎから、次式によりｓｉｇｎを計算する（ステップＳ２１２３ａ）。

【０１２２】

【数３９】

$$(sign \cdot s) \mod n$$

検証鍵識別子ｋｉｄに対応する署名値が無い場合は、データキャリア１の署名値部に新しいレコードを追加し（ステップＳ２１２２ｂ）、ステップＳ２１１９でセットしたｓをｓｉｇｎにセットする（ステップＳ２１２３ｂ）。データキャリア１のデータ部に上記ステップで生成された新しいレコードを追加し（ステップＳ２１２４）、ステップＳ２１２３ａまたはステップＳ２１２３ｂで計算された変数ｓｉｇｎの値でデータキャリア１の署名値部の検証鍵識別子ｋｉｄに対応する署名値を更新する（ステップＳ２１２５）。

【０１２３】

本実施例では署名鍵情報は既に署名鍵情報記憶手段１５に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップＳ２１１１で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップＳ２１１７は署名手段１６で、ステップＳ２１１８は物流情報生成手段１１で行うが、ステップＳ２１１８も署名手段１６で行うようにしても良いし、ステップＳ２１１７とステップＳ２１１８は署名手段１６と物流情報生成手段１１とで平行して計算するようにしてもよい。

【０１２４】

［実施例９］

図２２は、図１９に示される物流情報を検証するために物流情報検証部６で行われる検証処理の流れを示している。図２２において、読み取り手段４によりデータキャリア１から物流情報を読み取り（ステップＳ２２０１）、読み取った物流情報からデータ部を読み出し（ステップＳ２２０２）、変数ｒｉｄ，ｄａｔａをクリアする（ステップＳ２２０３）。ステップＳ２２０２で読み出したデータ部から１レコード読み出し（ステップＳ２２０４）、読み出したレコードから被署名部を読み出し、変数ｄにセットする（ステップＳ２２０５）。ステップＳ２２０５で読み出した被署名部から署名者識別子を読み出し、変数ｓｉｄにセットする（ステップＳ２２０６）。変数ｒｉｄに値がセットされているか確認し（ステップＳ２２０７）、セットされていなければ、ステップＳ２２０９に進み、セットされていれば、ステップＳ２２０６で読み出した署名者識別子ｓｉｄとｒｉ

ｄを比較する（ステップＳ２２０８）。ｓｉｄとｒｉｄが等しくなければ、レコードが不連続である旨を返し、終了し、等しければステップＳ２２０９に進む。ステップＳ２２０５で読み出した被署名部ｄと変数ｄａｔａの値を使って次式を計算し、変数ｄａｔａにセットする（ステップＳ２２０９）。

【０１２５】

【数４０】

ｄａｔａ｜ｄ　（｜はビット列の連結。ただしｄａｔａがクリアされている時は、ｄを返す）。

ここでステップＳ２２０９での計算は他の方法でもよく、ｄａｔａとｄを使った計算結果がｄａｔａとｄの組に対して一対一に対応していればよい。ステップＳ２２０２で読み出したデータ部に未処理のレコードがあるか確認し（ステップＳ２２１０）、未処理のレコードがあれば、ステップＳ２２０５で読み出した被署名部から受領者識別子を変数ｒｉｄにセットし（ステップＳ２２１１）、ステップＳ２２０４に戻り、次の未処理レコードの処理をおこない、未処理のレコードが無ければ、ステップＳ２２１２に進む。ステップＳ２２０５で読み出した被署名部ｄから製品識別子を読み出し、変数ｐｉｄにセットする（ステップＳ２２１２）。上記ステップにより計算された変数ｄａｔａ、ステップＳ２２０６で読み出した署名者識別子ｓｉｄ、ステップＳ２２１２で読み出した製品識別子ｐｉｄを使いハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算し、変数ｈにセットする（ステップＳ２２１３）。ここでハッシュ関数Ｈ（）は引数ｄａｔａ、ｐｉｄ、ｓｉｄから一意の値を返す関数Ｆ（ｄａｔａ，ｐｉｄ，ｓｉｄ）、例えば、ｄａｔａ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）、の返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであり、関数Ｆ（）、ハッシュ関数は特定のものに限定されない。ステップＳ２２０１で読み出した物流情報から署名値を読み出し、変数ｓｉｇｎにセットする（ステップＳ２２１４）。ステップＳ２２０５で読み出した被署名部ｄから検証鍵識別子を読み出し、変数ｋｉｄにセットする（ステップＳ２２１５）。一の検証鍵記憶手段１０からステップＳ２２１５で読み出した第検証鍵識別子ｋｉｄに対応する署名検証鍵を読み出し、変数Ｅ，ｎにセットする（ステップＳ２２１６）。ステップＳ２２１４で読み出した

署名値ｓｉｇｎとステップＳ２２１６で読み出した署名検証鍵Ｅ，ｎを使い次式により検証用の値を計算し、変数ｖａｌにセットする（ステップＳ２２１７）。

【０１２６】

【数４１】

$$sign^{E} \mod n$$

ステップＳ２２１３で計算したハッシュ値ｈとステップＳ２２１７で計算した検証用の値ｖａｌとを比較する（ステップＳ２２１８）。ｈとｖａｌが等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【０１２７】

図２３は、図１９に示される物流情報を生成するために物流情報生成部７で行われる処理の流れを示している。図２３において、変数ｄａｔａを初期化し（ステップＳ２３０１）、データキャリア１内に物流情報があるか確認し（ステップＳ２３０２）、無ければ、物流情報のデータ部に記載するための新しいレコードの製品識別子ｐｉｄをセットし（ステップＳ２３０３ａ）、新しいレコードの署名者識別子ｓｉｄをセットし（ステップＳ２３０４ａ）、ステップＳ２３１３へ進む。ステップＳ２３０２で物流情報があれば、データキャリア１から読み取り手段４を介し物流情報を読み出し（ステップＳ２３０３ｂ）、読み出した物流情報からデータ部を取出し（ステップＳ２３０４ｂ）、そのデータ部から１レコード取出す（ステップＳ２３０５ｂ）。取出したレコードの被署名部を取出し、変数ｄにセットする（ステップＳ２３０６ｂ）。変数ｄａｔａとステップＳ２３０６ｂで取出した被署名部ｄとから次式の計算を行い、変数ｄａｔａにセットする（ステップＳ２３０７ｂ）。

【０１２８】

【数４２】

ｄａｔａ｜ｄ　（｜はビット列の連結。ただしｄａｔａがクリアされている時は、ｄを返す）

ステップＳ２３０７ｂの計算は、他の方式でも良いが、物流情報検証部６で行われる検証処理での計算（図２２のステップＳ２２０９）と同じであればよい。ス

テップＳ２３０４ｂで読み出したデータ部に未処理のレコードがあるか確認し（ステップＳ２３０８ｂ）、未処理のレコードが無ければステップＳ２３０９ｂに進み、未処理のレコードがあれば、ステップＳ２３０５ｂに戻り、次のレコードを処理する。上記ステップにより読み出された被署名部ｄから製品識別子を読み出し、新しいレコードの製品識別子ｐｉｄにセットし（ステップＳ２３０９ｂ）、被署名部ｂから受領者識別子ｒｉｄを読み出す（ステップＳ２３１０ｂ）。署名者秘密情報保持手段１７から署名者識別子を読み出し、新しいレコードの署名者識別子ｓｉｄにセットする（ステップＳ２３１１ｂ）。ステップＳ２３１０ｂで読み出した受領者識別子ｒｉｄとステップＳ２３１１ｂで読み出した署名者識別子ｓｉｄを比較し（ステップＳ２３１２ｂ）、受領者識別子ｒｉｄと署名者識別子ｓｉｄが等しければ、ステップＳ２３１３へ進み、等しくなければ受領者と署名者が一致しない旨を返し、終了する。製品の受領者の識別子を新しいレコードの受領者識別子ｒｉｄにセットする（ステップＳ２３１３）。被署名部ｄから署名検証鍵識別子を読み出し、新しいレコードの署名検証鍵識別子ｋｉｄにセットする（ステップＳ２３１４）。新しいレコードの残りのフィールドをセットし（ステップＳ２３１５）、新しいレコードの被署名部を取り出し、変数ｄへセットする（ステップＳ２３１６）。変数ｄａｔａと変数ｄに対し、ステップＳ２３０７ｂと同じ計算を行い、変数ｄａｔａにセットする（ステップＳ２３１７）。ステップＳ２３０３ａまたはステップＳ２３０９ｂでセットした製品識別子ｐｉｄと、ステップＳ２３０４ａまたはステップＳ２３１１ｂでセットした署名者識別子ｓｉｄと、ステップＳ２３１７でセットした変数ｄａｔａに対し、ハッシュ値Ｈ（ｄａｔａ，ｐｉｄ，ｓｉｄ）を計算する（ステップＳ２３１８）。ここでハッシュ関数Ｈ（）は、物流情報検証部６で行われる検証処理（図２２のステップＳ２３１３）で用いられるものと同じである。署名者秘密情報記憶手段１７から署名者秘密情報を読み出し、変数ｄにセットし（ステップＳ２３１９）、署名鍵情報記憶手段１４から製品識別子ｐｉｄに対応する署名鍵情報を取出し、変数ｔ，ｎにセットする（ステップＳ２３２０）。次式により、第一の署名値を計算し、変数ｓ１にセットする（ステップＳ２３２１）。

【０１２９】

【数４３】

$$h^{f(d,n,pid,sid)} \quad mod \quad n$$

ここで、関数ｆ（）は引数ｄ、ｎ、ｐｉｄ、ｓｉｄから一意の値を返し、その値からｄ、ｎ、ｐｉｄ、ｓｉｄの値を知ることができない一方向性を持つ関数である。例えば、ｄ｜ｎ｜ｐｉｄ｜ｓｉｄ（｜はビット列の連結）に対して返す値に対してＳＨＡ－１、ＭＤ５などハッシュ関数を適用したものであるが、これに限定されない。次式により、第二の署名値を計算し、変数ｓ２にセットする（ステップＳ２３２２）。

【０１３０】

【数４４】

$$h^{t} \quad mod \quad n$$

ステップＳ２３２０で計算された第一の署名値ｓ１、ステップＳ２３２２で計算された第二の署名値ｓ２を使い次式により署名値を計算し、変数ｓｉｇｎにセットする（ステップＳ２３２３）。

【０１３１】

【数４５】

$$(s1 \cdot s2) \quad mod \quad n$$

書き込み手段５を介し、データキャリア１内のデータ部に新しいレコードを追加し（ステップＳ２３２４）、データキャリア１内の署名値をステップＳ２３２３で計算した変数ｓｉｇｎの値で更新する（ステップＳ２３２５）。

【０１３２】

［実施例１０］

図２４は、図１の物流情報処理モジュール２および物流情報管理モジュール３をそれぞれ部分モジュールに分離した実施例を示している。図２４において、物流情報処理モジュール２（図１）を第一の検証モジュール３２および書き込みモジュール３７に分離している。また、物流情報管理モジュール３（図１）を第二の検証モジュール３３および署名鍵情報生成モジュール３４に分離している。第一の検証モジュール３２および書き込みモジュール３７にはそれぞれ第１の読み取り手段３５および第２の読み取り手段３６が設けられている。また署名鍵情報

生成モジュール３４には第三の通信手段２８が設けられ、第二の検証モジュール３３と通信を行うようになっている。なお、図２４において図１または図６と対応する個所には対応する符号を付して説明を繰り返さない。

【０１３３】

【発明の効果】

以上で説明したように、本発明によれば、流通業者が認証局から証明書を取得する必要が無く、また秘密情報がＩＣカードなどの耐タンパな容器に収められるためセキュリティ確保が容易であるので、多大な設備が必要なく管理が容易な流通管理システムを実現することができる。また、本発明において、署名回数を制限するようにすれば、偽造商品に本物の商品と同一のＩＤを振り、署名をして別の受け取り業者へ配送するといった不正を防ぐことができる。

【図面の簡単な説明】

【図１】　本発明の実施例１を用いて本発明の基本の構成を説明するブロック図である。

【図２】　実施例１の物流情報のデータ構造を説明する図である。

【図３】　実施例１の検証時の流れを説明するフローチャートである。

【図４】　実施例１の署名時の流れを説明するフローチャートである。

【図５】　実施例１の署名鍵情報取得時の流れを説明するフローチャートである。

【図６】　本発明の実施例２の情報検証部の構成を説明するブロック図である。

【図７】　実施例２の物流情報のデータ構造を説明する図である。

【図８】　実施例２の検証時の流れを説明するフローチャートである。

【図９】　本発明の実施例３の署名モジュールの構成を説明するブロック図である。

【図１０】　実施例３の署名時の流れを説明するフローチャートである。

【図１１】　実施例４における署名モジュールの署名時の流れを説明するフローチャートである。

【図１２】　本発明の実施例５の署名鍵情報生成モジュールの構成を説明す

るブロック図である。

【図１３】　実施例５の署名鍵情報取得時の流れを説明するフローチャートである。

【図１４】　本発明の実施例６の物流情報のデータ構造を説明する図である。

【図１５】　実施例６の検証時の流れを説明するフローチャートである。

【図１６】　実施例６の署名時の流れを説明するフローチャートである。

【図１７】　実施例７の物流情報検証部における検証時の流れを説明するフローチャートである。

【図１８】　実施例７の署名時の流れを説明するフローチャートである。

【図１９】　本発明の実施例８の物流情報のデータ構造を説明する図である。

【図２０】　実施例８の検証時の流れを説明するフローチャートである。

【図２１】　実施例８の署名時の流れを説明するフローチャートである。

【図２２】　本発明の実施例９の検証時の流れを説明するフローチャートである。

【図２３】　実施例９の署名時の流れを説明するフローチャートである。

【図２４】　本発明の実施例１０の構成を説明するブロック図である。

【符号の説明】

１　　データキャリア

２　　物流情報処理モジュール

３　　物流情報管理モジュール

４　　読み出し手段

５　　書き込み手段

６　　第一の情報検証部

７　　情報生成部

８　　第一の通信手段

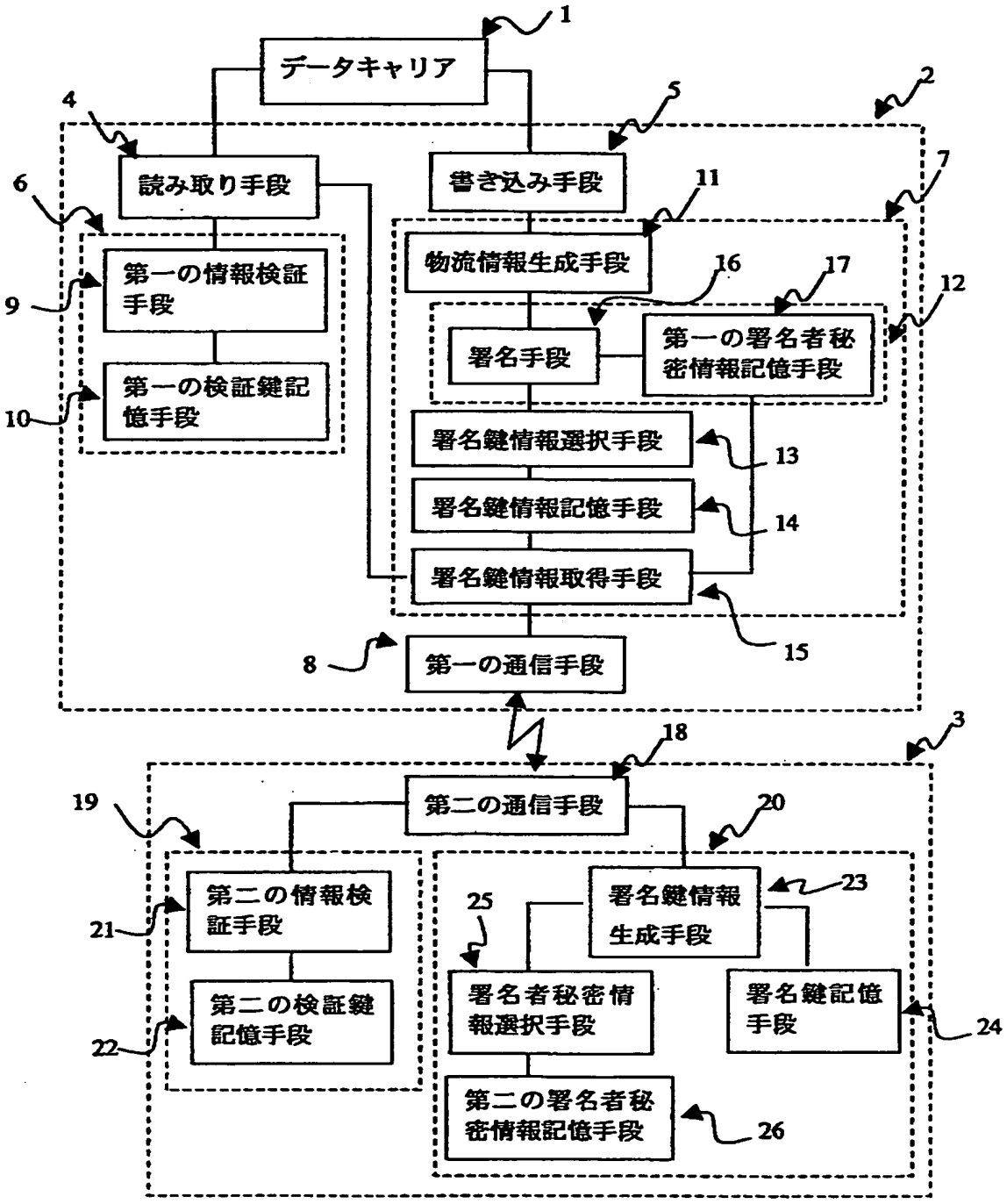
９　　第一の情報検証手段

１０　　第一の検証鍵記憶手段

１１　物流情報生成手段
１２　署名モジュール
１３　署名鍵情報選択手段
１４　署名鍵情報記憶手段
１５　署名鍵情報取得手段
１６　署名手段
１７　第一の署名者秘密情報記憶手段
１８　第二の通信手段
１９　第二の情報検証部
２０　署名鍵情報生成部
２１　第二の情報検証手段
２２　第二の検証鍵記憶手段
２３　署名鍵情報生成手段
２４　署名鍵記憶手段
２５　署名者秘密情報選択手段
２６　第二の署名者秘密情報記憶手段
２７　署名鍵使用限度情報保持手段
２８　第三の通信手段
２９　第一の検証鍵選択手段
３０　第二の検証鍵選択手段
３１　署名鍵選択手段
３２　第一の検証モジュール
３３　第二の検証モジュール
３４　署名鍵情報情報生成モジュール
３５　第一の読み出し手段
３６　第二の読み出し手段

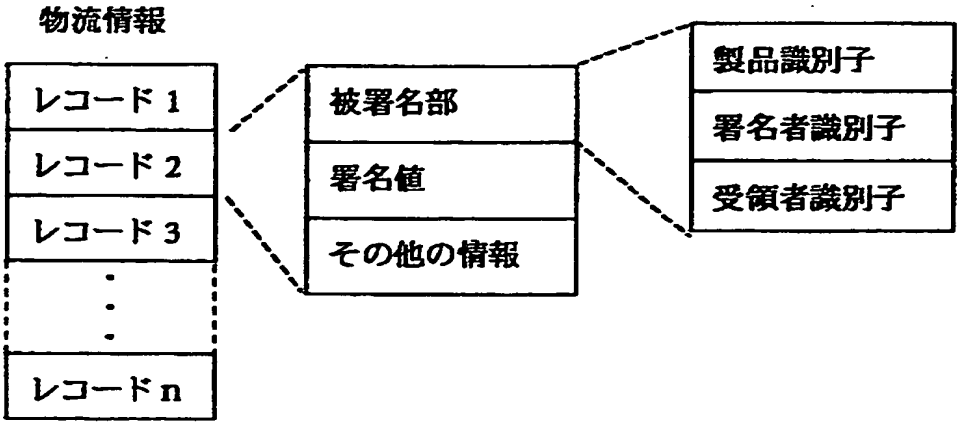
【書類名】　　図面

【図１】



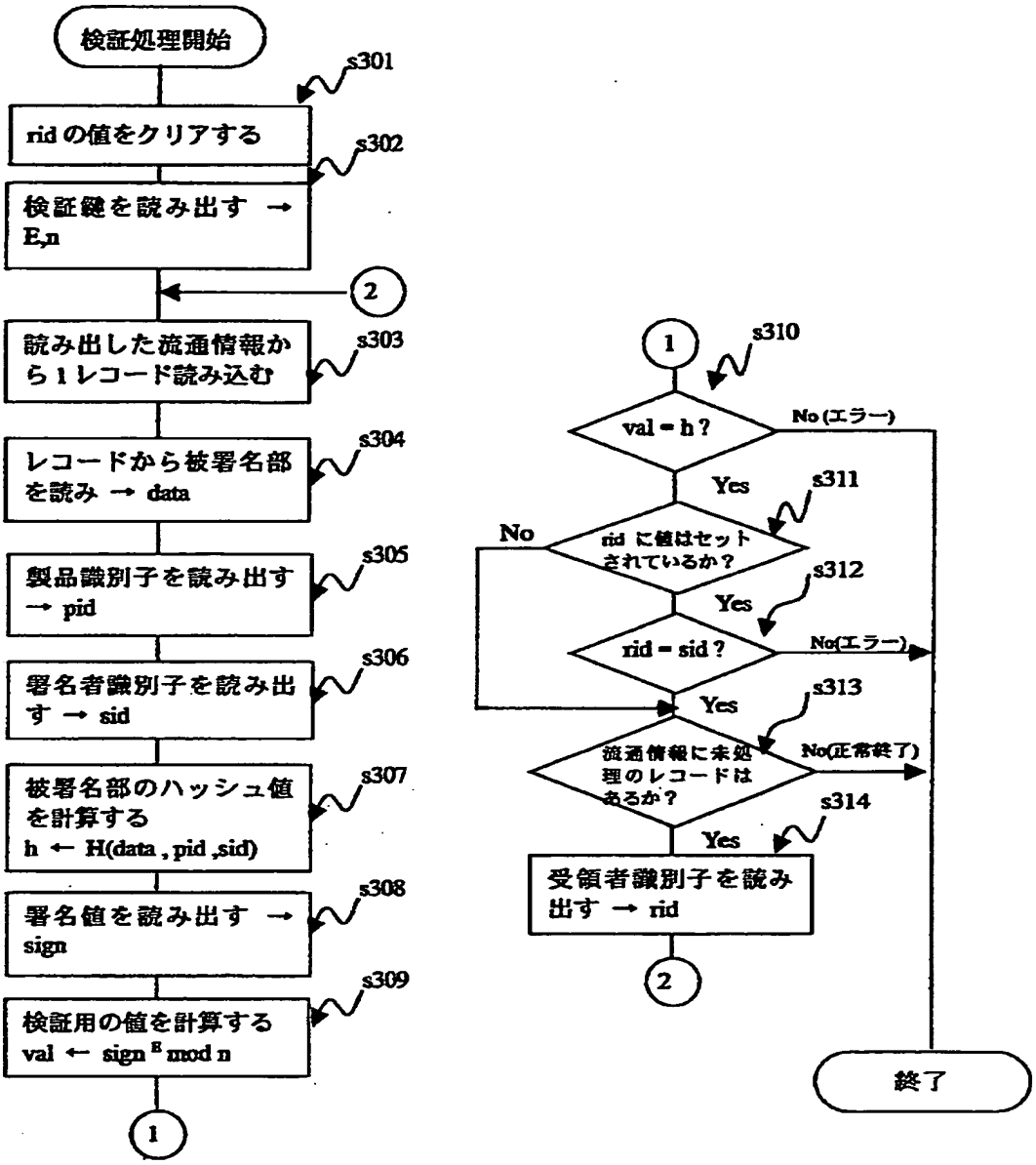
基本の構成

【図２】



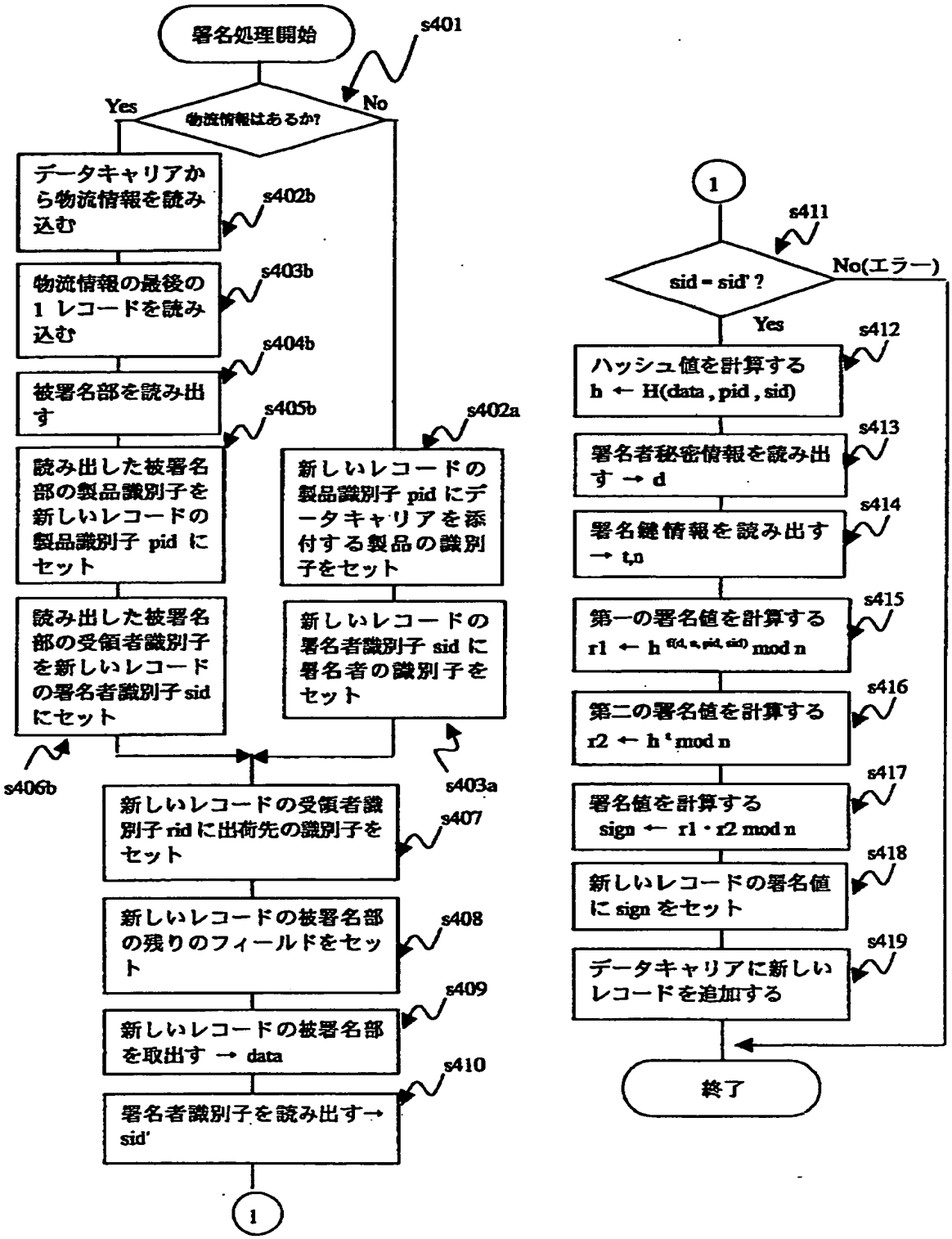
データキャリアに記憶される物流情報

【図３】


検証処理開始
s301
rid の値をクリアする
s302
検証鍵を読み出す → E,n
2
読み出した流通情報から１レコード読み込む
s303
レコードから被署名部を読み → data
s304
製品識別子を読み出す → pid
s305
署名者識別子を読み出す → sid
s306
被署名部のハッシュ値を計算する
h ← H(data , pid ,sid)
s307
署名値を読み出す → sign
s308
検証用の値を計算する
val ← sign^E mod n
s309
1
1
s310
val = h ?
No (エラー)
Yes
s311
No
rid に値はセットされているか？
Yes
s312
rid = sid ?
No(エラー)
Yes
s313
流通情報に未処理のレコードはあるか？
No(正常終了)
Yes
s314
受領者識別子を読み出す → rid
2
終了

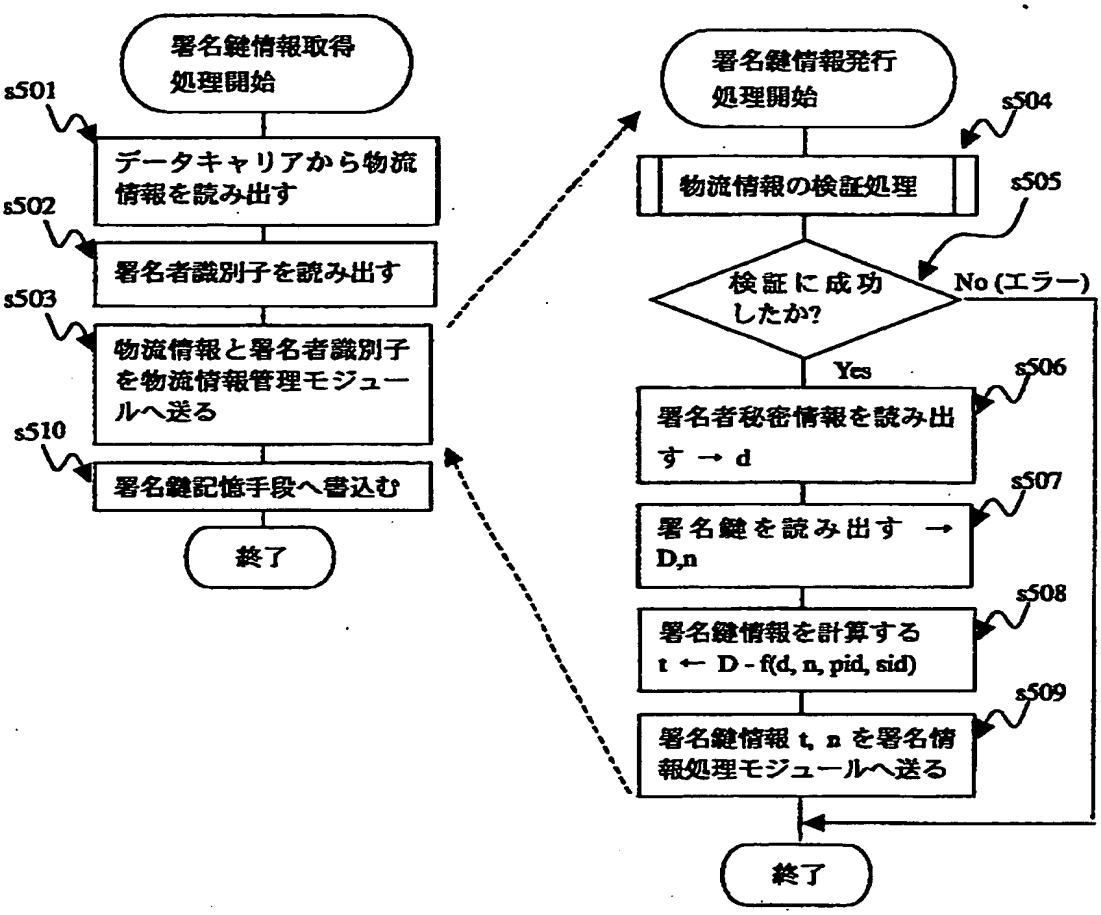
検証時の処理

【図４】


署名処理開始
s401
物流情報はあるか?
Yes
No
データキャリアから物流情報を読み込む
s402b
物流情報の最後の1レコードを読み込む
s403b
被署名部を読み出す
s404b
読み出した被署名部の製品識別子を新しいレコードの製品識別子 pid にセット
s405b
読み出した被署名部の受領者識別子を新しいレコードの署名者識別子 sid にセット
s406b
新しいレコードの製品識別子 pid にデータキャリアを添付する製品の識別子をセット
s402a
新しいレコードの署名者識別子 sid に署名者の識別子をセット
s403a
新しいレコードの受領者識別子 rid に出荷先の識別子をセット
s407
新しいレコードの被署名部の残りのフィールドをセット
s408
新しいレコードの被署名部を取出す → data
s409
署名者識別子を読み出す→ sid'
s410
1
1
s411
sid = sid' ?
No(エラー)
Yes
ハッシュ値を計算する h ← H(data , pid , sid)
s412
署名者秘密情報を読み出す → d
s413
署名鍵情報を読み出す → t,n
s414
第一の署名値を計算する r1 ← h^f(d, t, pid, sid) mod n
s415
第二の署名値を計算する r2 ← h^t mod n
s416
署名値を計算する sign ← r1・r2 mod n
s417
新しいレコードの署名値に sign をセット
s418
データキャリアに新しいレコードを追加する
s419
終了

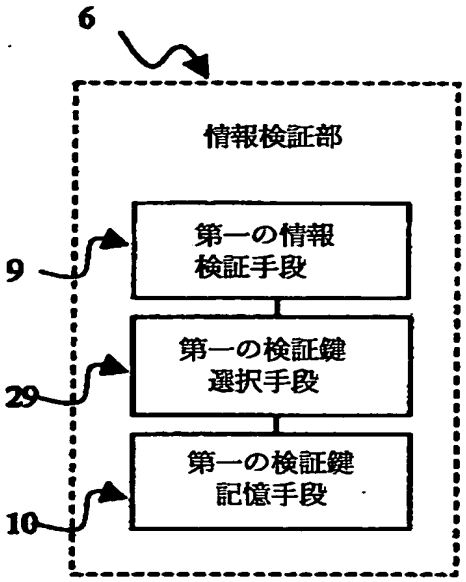
署名時の処理

【図５】



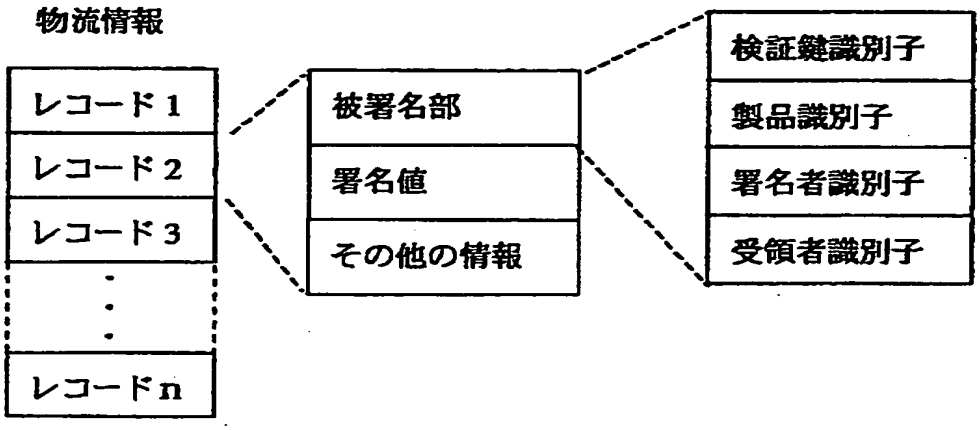
署名鍵情報取得時の処理

【図６】



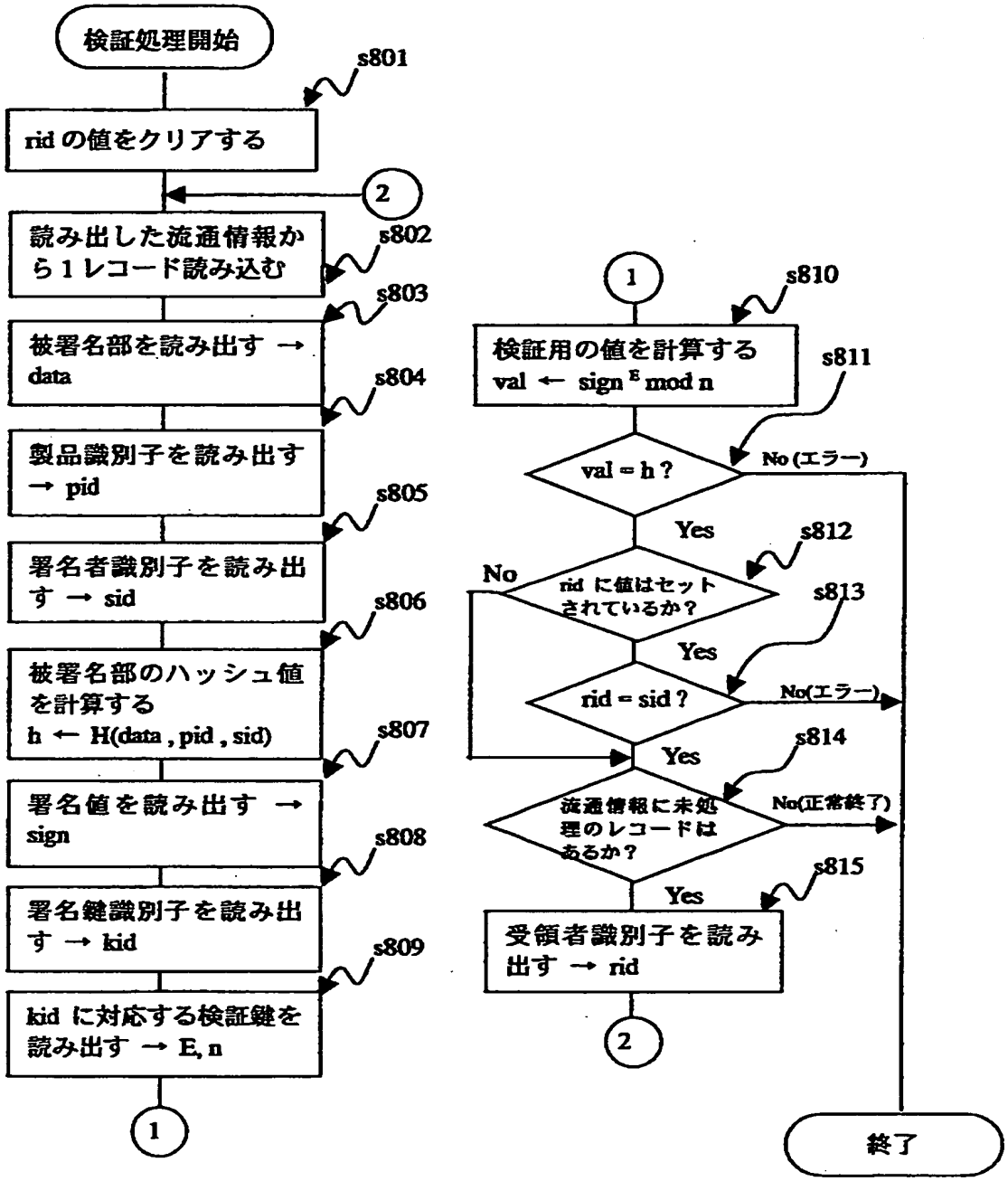
情報検証部

【図７】



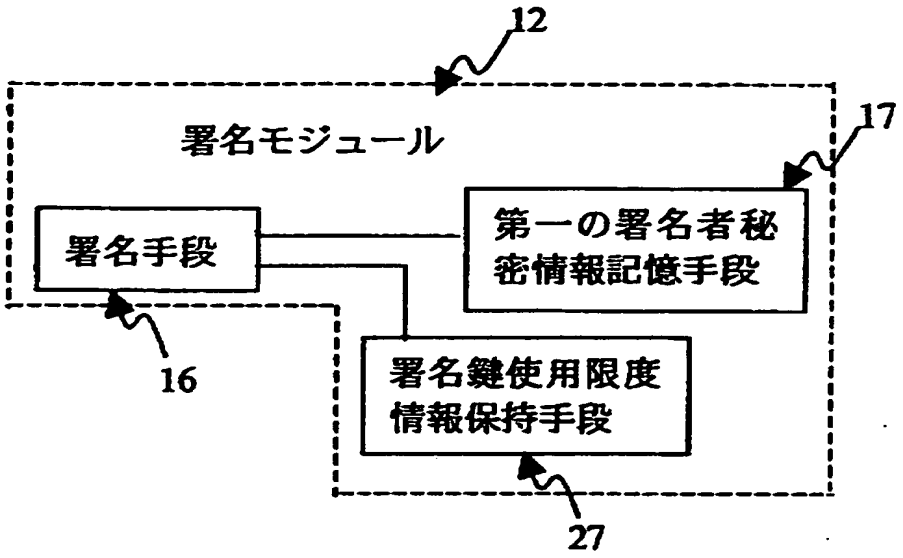
データキャリアに記憶される物流情報

【図８】


検証処理開始
s801
rid の値をクリアする
2
読み出した流通情報から１レコード読み込む
s802
s803
被署名部を読み出す → data
s804
製品識別子を読み出す → pid
s805
署名者識別子を読み出す → sid
s806
被署名部のハッシュ値を計算する
h ← H(data , pid , sid)
s807
署名値を読み出す → sign
s808
署名鍵識別子を読み出す → kid
s809
kid に対応する検証鍵を読み出す → E, n
1
1
s810
検証用の値を計算する
val ← sign E mod n
s811
val = h ?
No (エラー)
Yes
s812
No
rid に値はセットされているか？
s813
Yes
rid = sid ?
No(エラー)
Yes
s814
流通情報に未処理のレコードはあるか？
No(正常終了)
Yes
s815
受領者識別子を読み出す → rid
2
終了

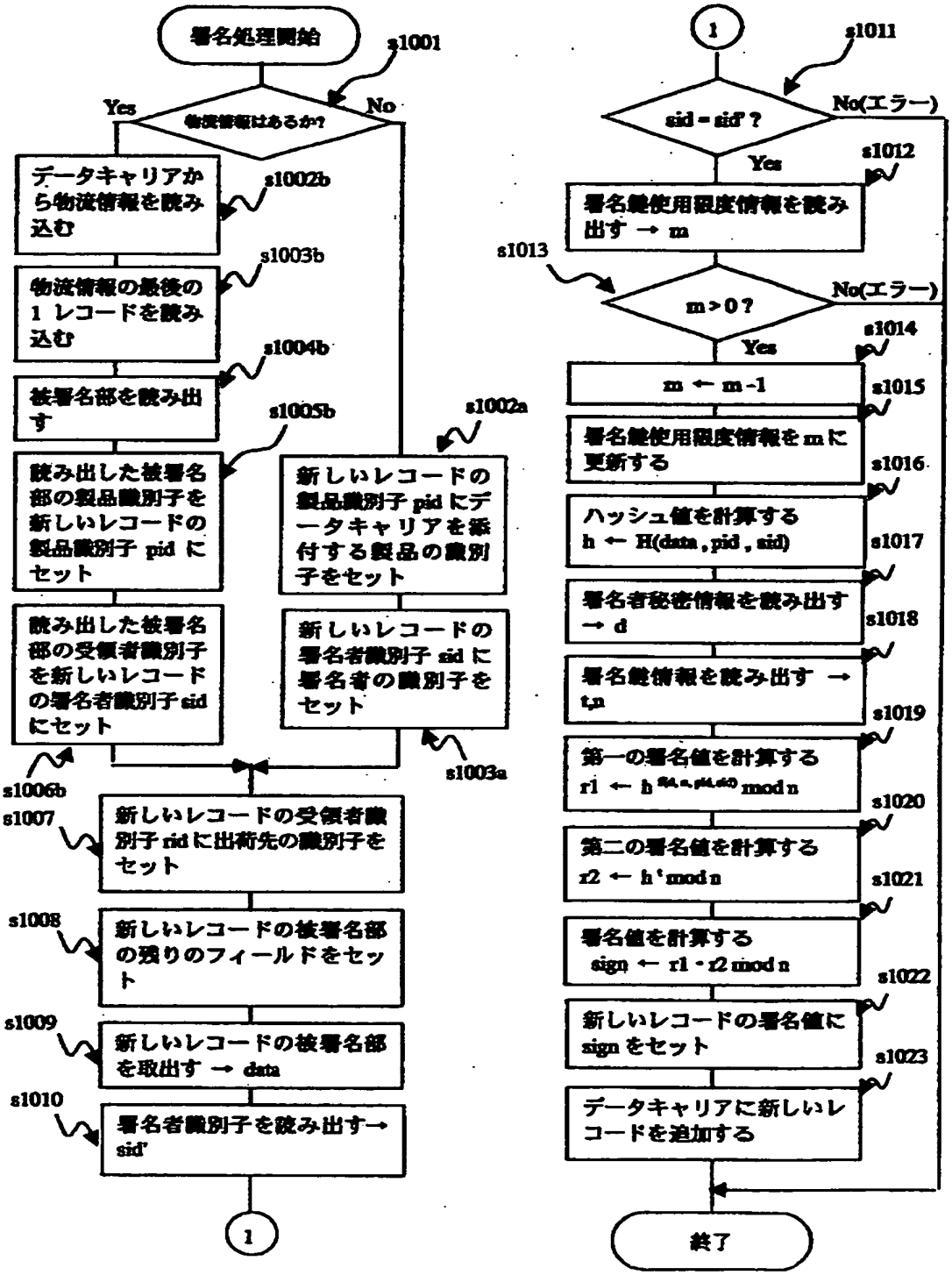
検証時の処理

【図９】



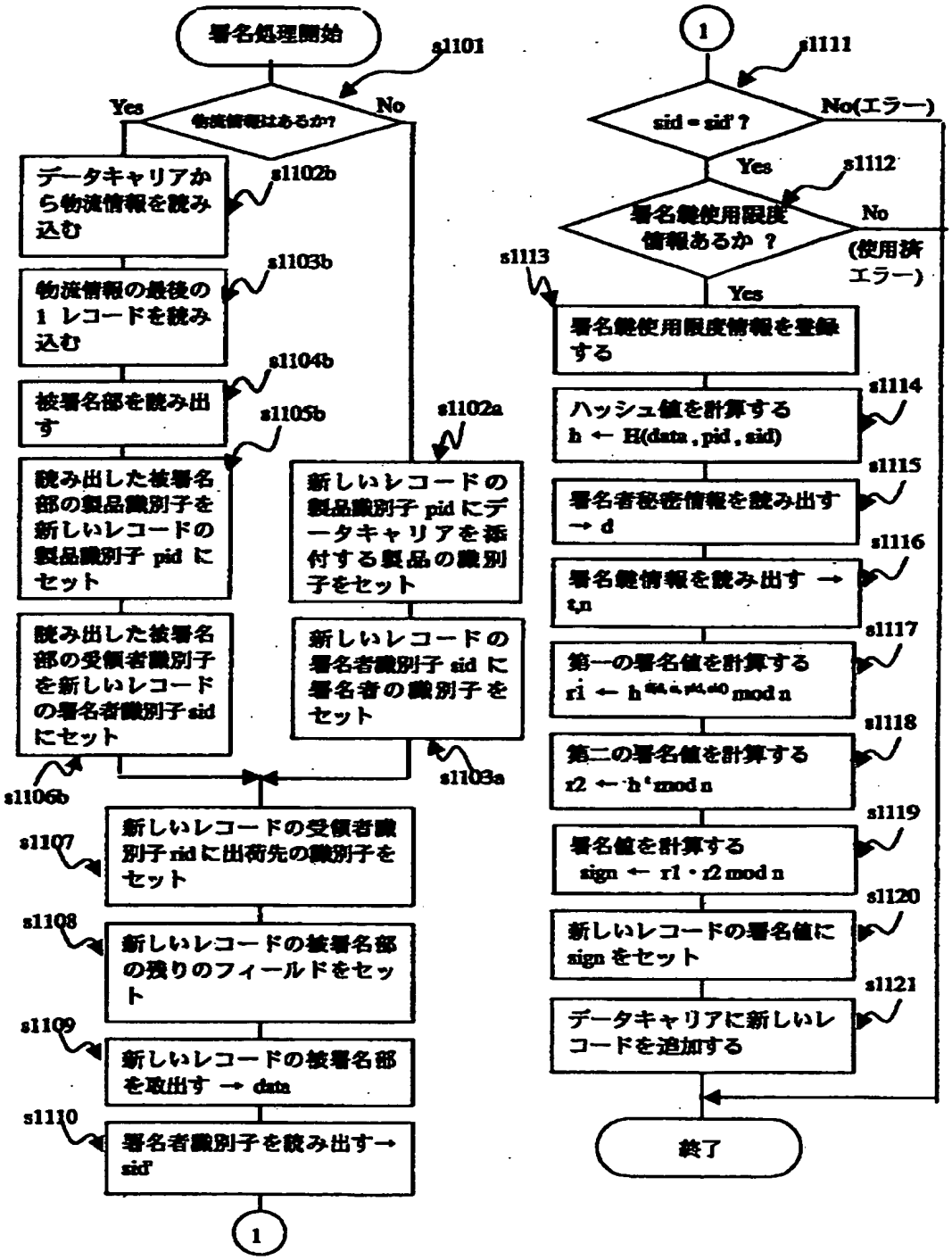
署名モジュール

【図１０】


署名処理開始
s1001
Yes
物流情報はあるか?
No
データキャリアから物流情報を読み込む
s1002b
物流情報の最後の1レコードを読み込む
s1003b
被署名部を読み出す
s1004b
読み出した被署名部の製品識別子を新しいレコードの製品識別子 pid にセット
s1005b
読み出した被署名部の受領者識別子を新しいレコードの署名者識別子 sid にセット
s1006b
新しいレコードの製品識別子 pid にデータキャリアを添付する製品の識別子をセット
s1002a
新しいレコードの署名者識別子 sid に署名者の識別子をセット
s1003a
新しいレコードの受領者識別子 rid に出荷先の識別子をセット
s1007
新しいレコードの被署名部の残りのフィールドをセット
s1008
新しいレコードの被署名部を取出す → data
s1009
署名者識別子を読み出す→ sid'
s1010
1
1
s1011
sid = sid' ?
No(エラー)
Yes
署名鍵使用限度情報を読み出す → m
s1012
m > 0 ?
s1013
No(エラー)
Yes
m ← m -1
s1014
署名鍵使用限度情報を m に更新する
s1015
ハッシュ値を計算する h ← H(data , pid , sid)
s1016
署名者秘密情報を読み出す → d
s1017
署名鍵情報を読み出す → t,n
s1018
第一の署名値を計算する r1 ← h^{d,n,pid,sid} mod n
s1019
第二の署名値を計算する r2 ← h' mod n
s1020
署名値を計算する sign ← r1 · r2 mod n
s1021
新しいレコードの署名値に sign をセット
s1022
データキャリアに新しいレコードを追加する
s1023
終了

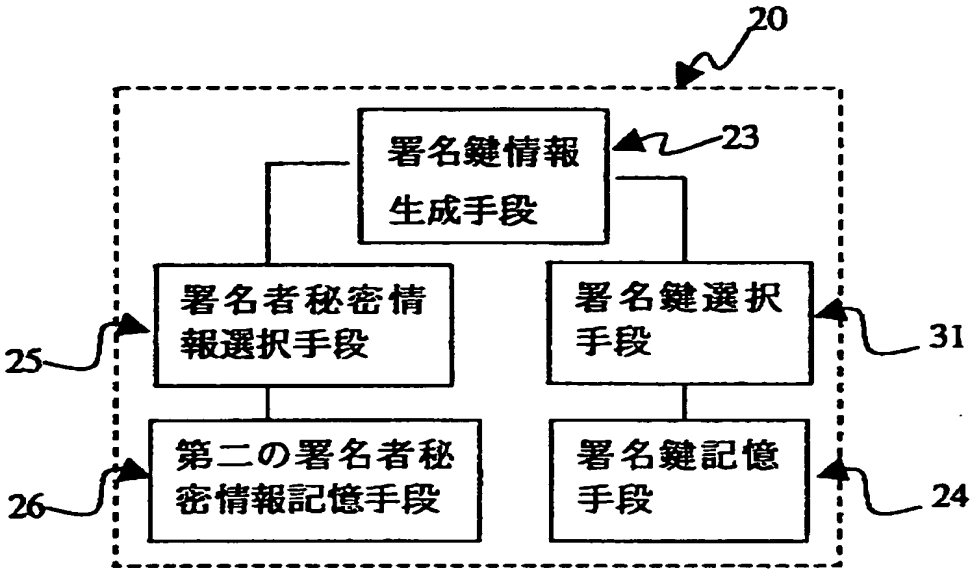
署名時の処理

【図１１】


署名処理開始
s1101
物流情報はあるか?
Yes
No
データキャリアから物流情報を読み込む
s1102b
物流情報の最後の1レコードを読み込む
s1103b
被署名部を読み出す
s1104b
読み出した被署名部の製品識別子を新しいレコードの製品識別子 pid にセット
s1105b
読み出した被署名部の受領者識別子を新しいレコードの署名者識別子sidにセット
s1106b
新しいレコードの製品識別子 pid にデータキャリアを添付する製品の識別子をセット
s1102a
新しいレコードの署名者識別子 sid に署名者の識別子をセット
s1103a
新しいレコードの受領者識別子 rid に出荷先の識別子をセット
s1107
新しいレコードの被署名部の残りのフィールドをセット
s1108
新しいレコードの被署名部を取出す → data
s1109
署名者識別子を読み出す→ sid'
s1110
1
1
sid = sid' ?
s1111
No(エラー)
Yes
署名鍵使用限度情報あるか ?
s1112
No
(使用済エラー)
Yes
署名鍵使用限度情報を登録する
s1113
ハッシュ値を計算する h ← H(data , pid , sid)
s1114
署名者秘密情報を読み出す → d
s1115
署名鍵情報を読み出す → t,n
s1116
第一の署名値を計算する r1 ← h^(d,k,pid,sid) mod n
s1117
第二の署名値を計算する r2 ← h^k mod n
s1118
署名値を計算する sign ← r1・r2 mod n
s1119
新しいレコードの署名値に sign をセット
s1120
データキャリアに新しいレコードを追加する
s1121
終了

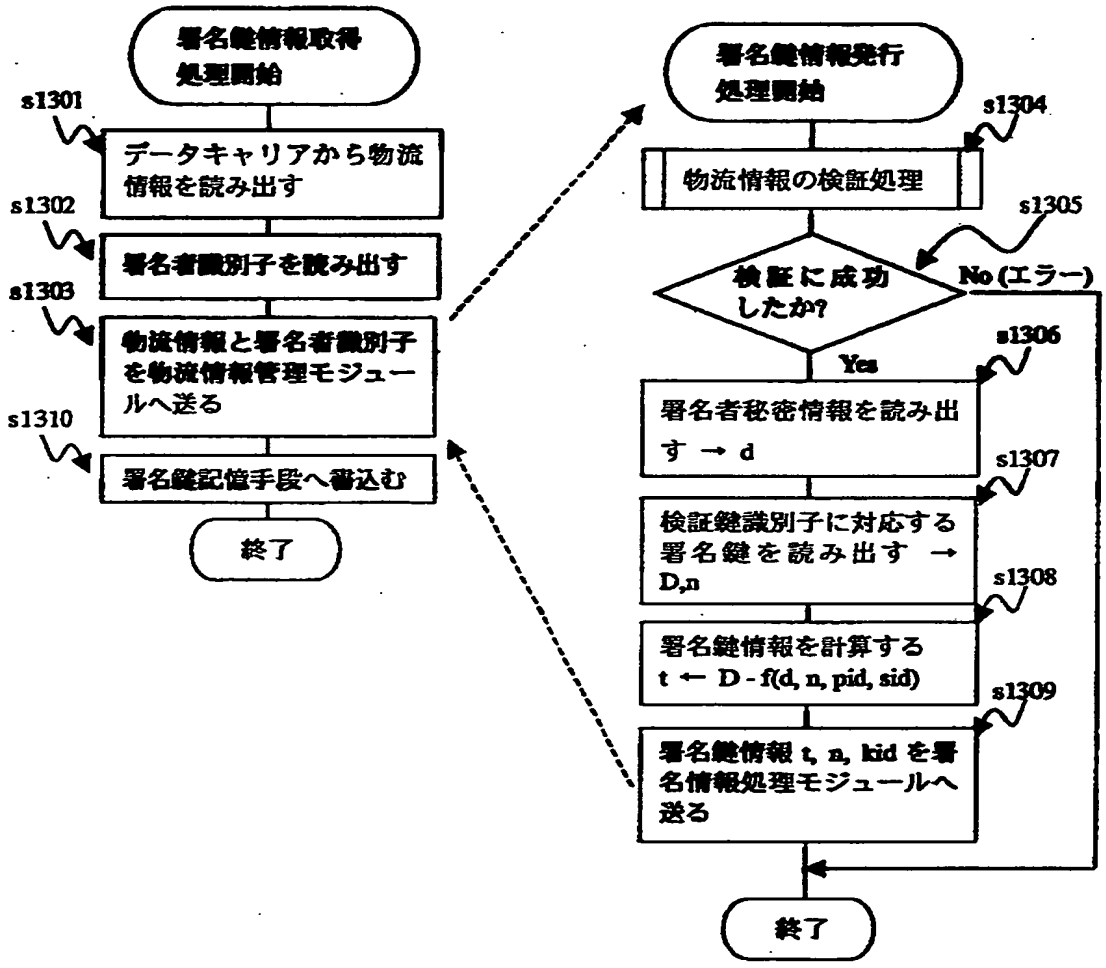
署名時の処理

【図１２】



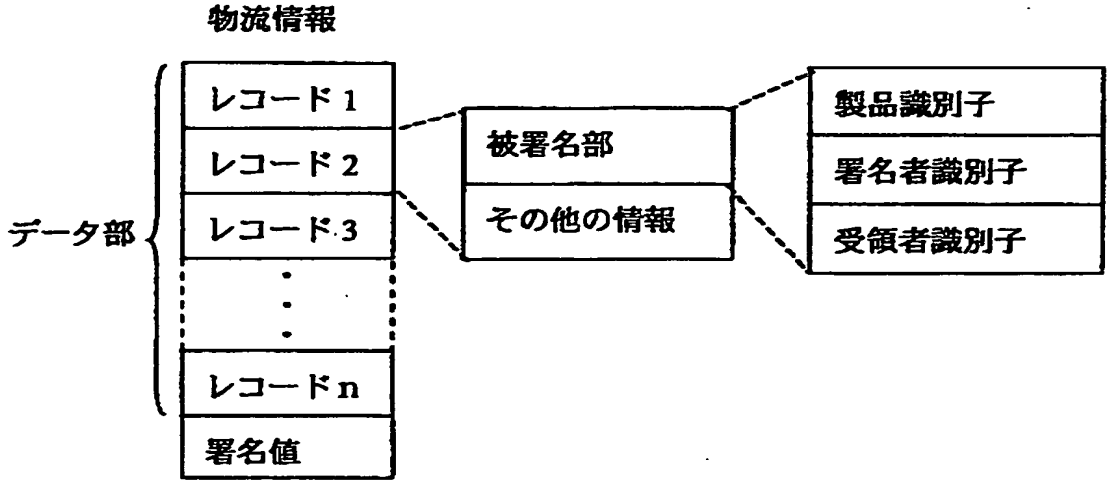
署名鍵情報情報生成モジュール

【図１３】



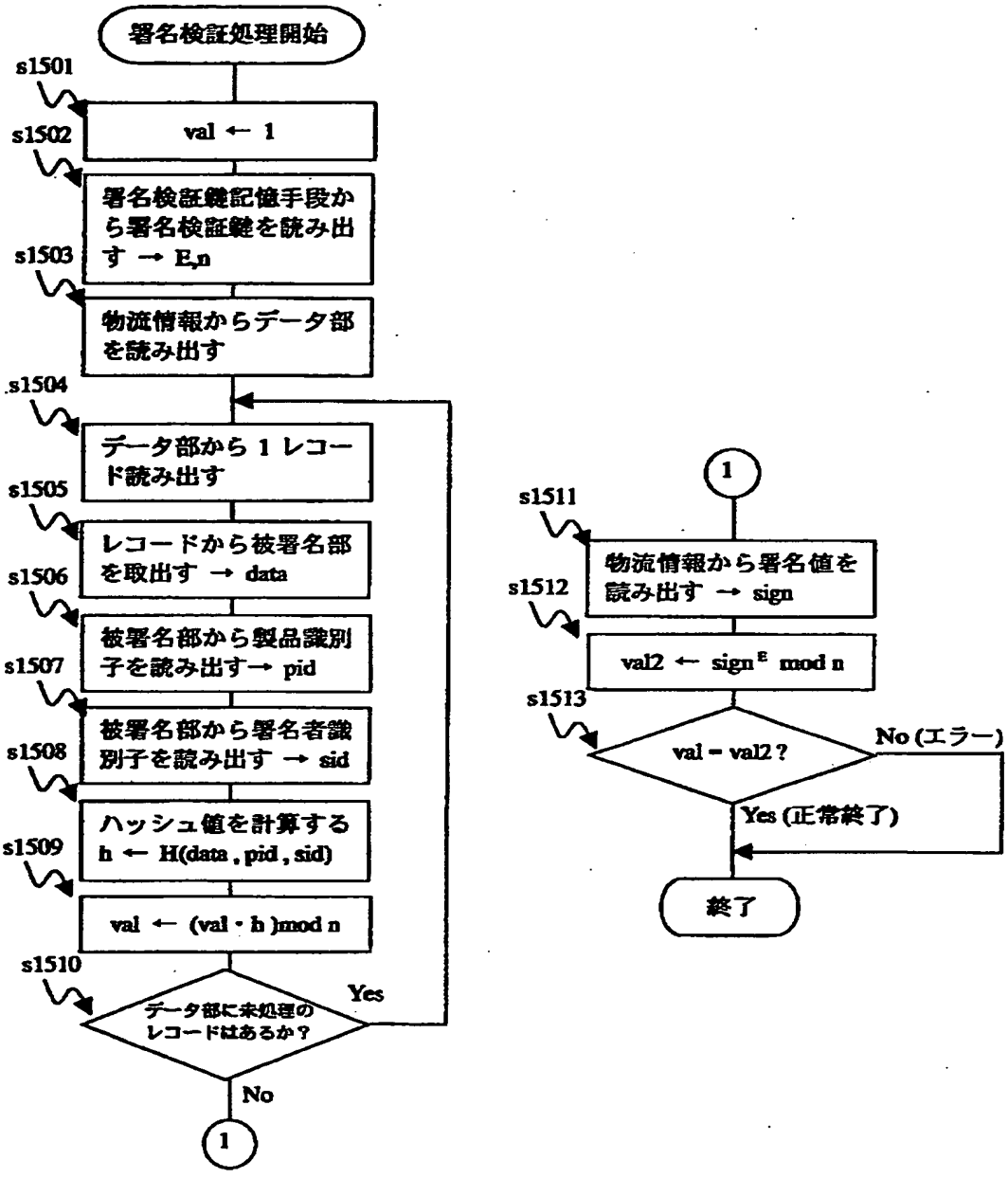
署名鍵情報取得時の処理

【図１４】



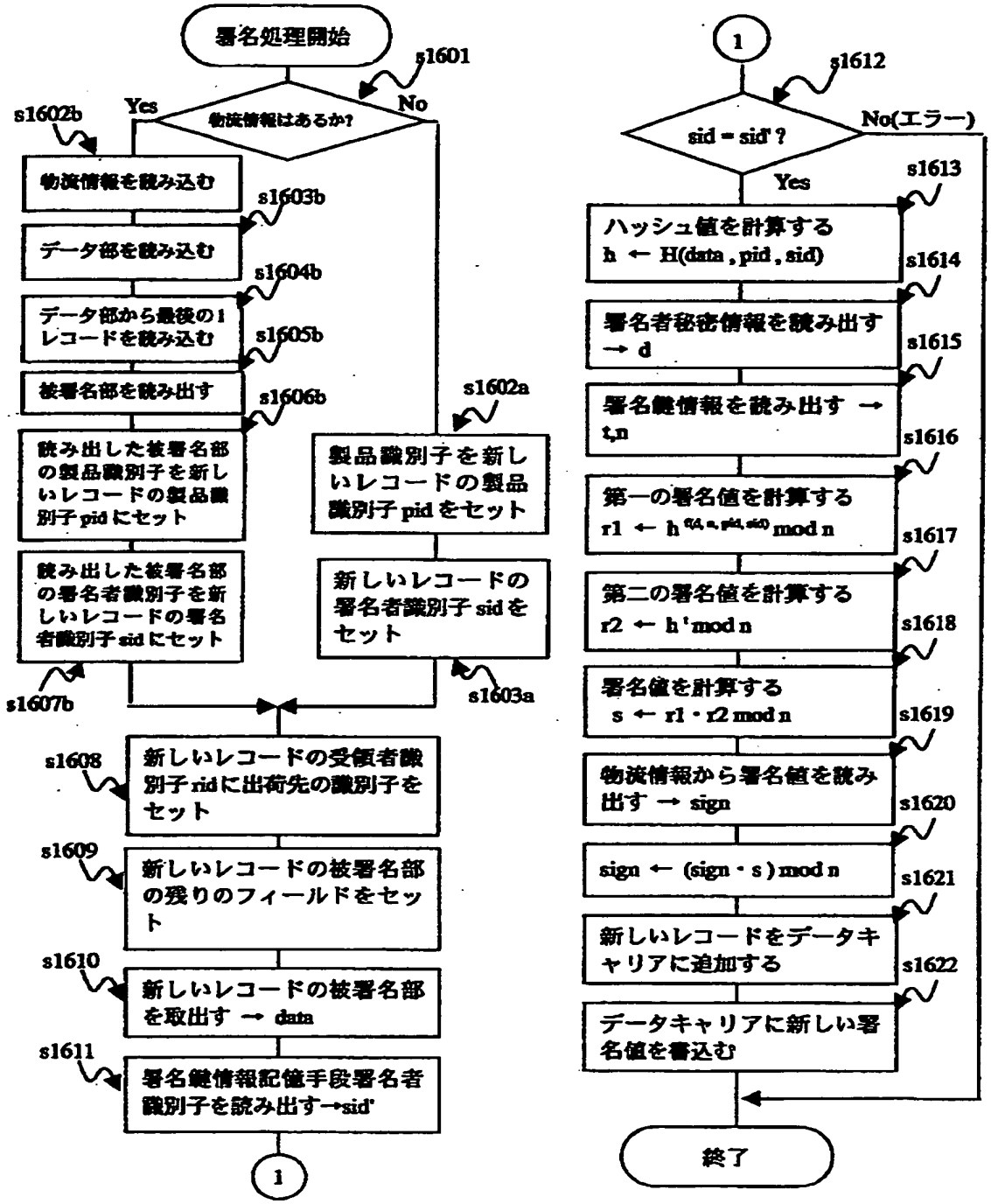
データキャリアに記憶される物流情報

【図１５】



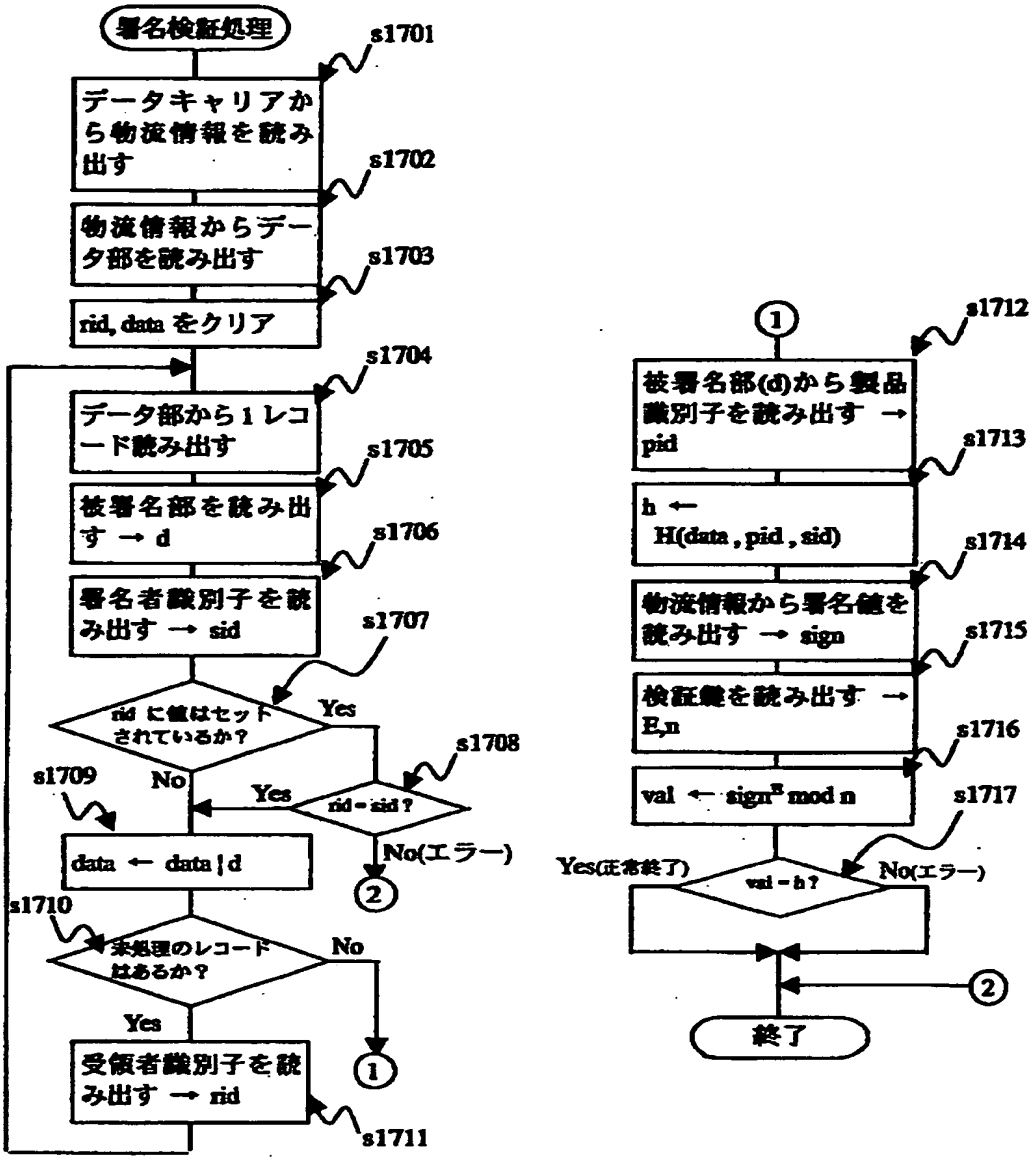
データキャリアに署名値を一つしか持たない場合の署名検証処理

【図１６】



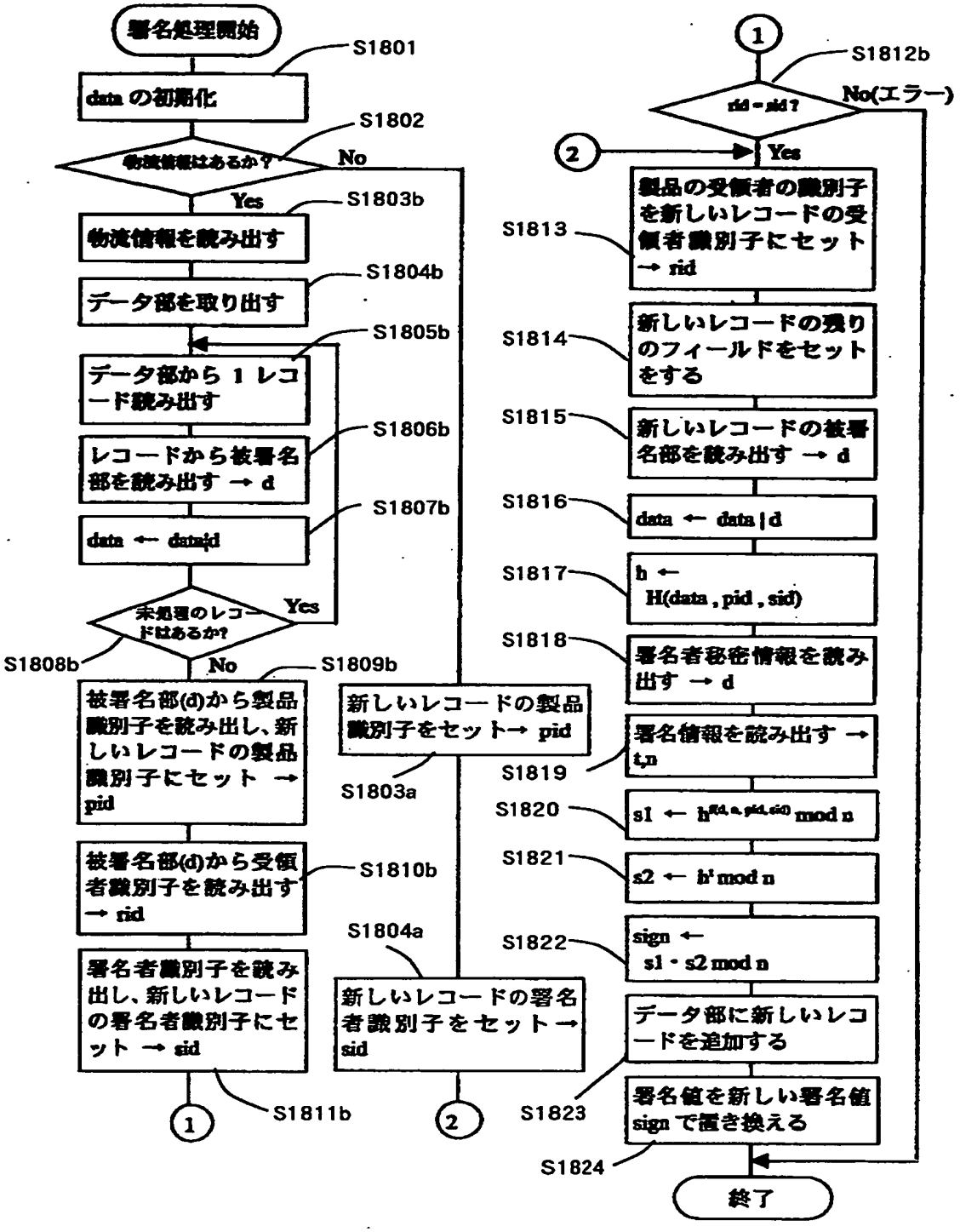
データキャリアに署名値を一つしか持たない場合の署名処理

【図１７】



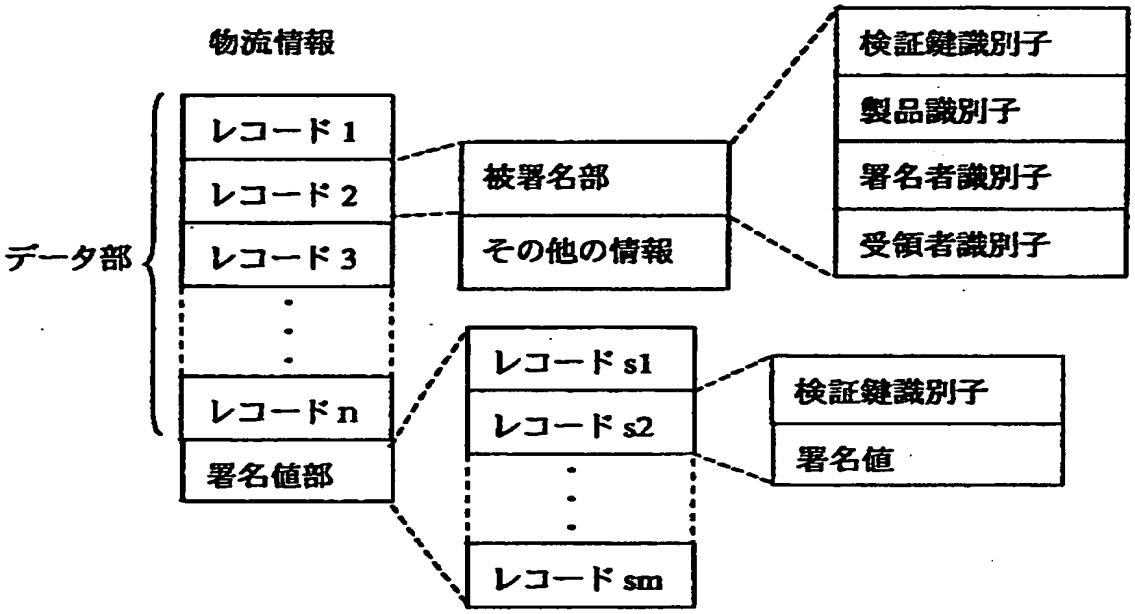
全体に署名をつける場合の検証処理（検証鍵は１つ）

【図１８】



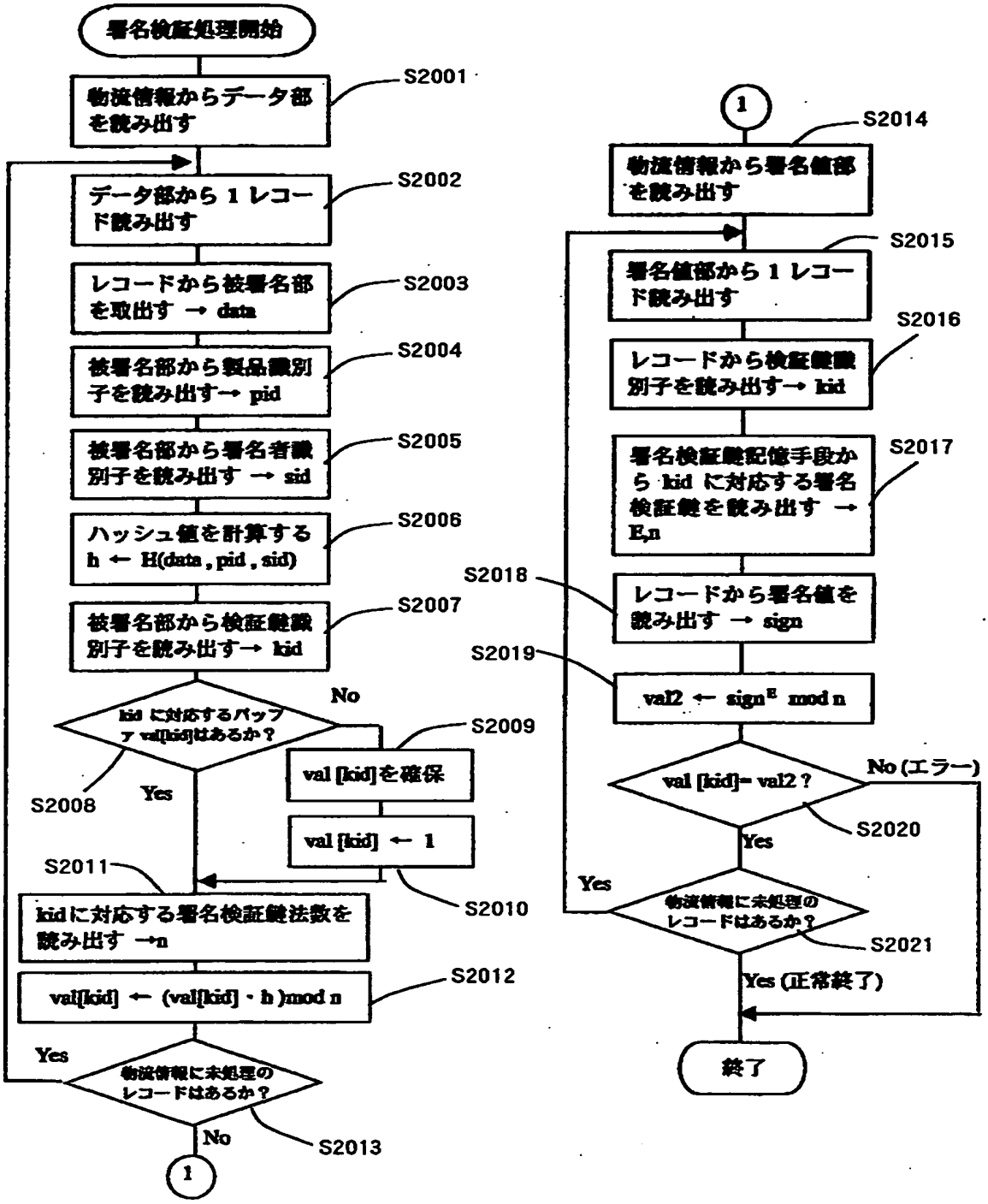
全体に署名をつける場合の署名処理（検証鍵は１つ）

【図１９】



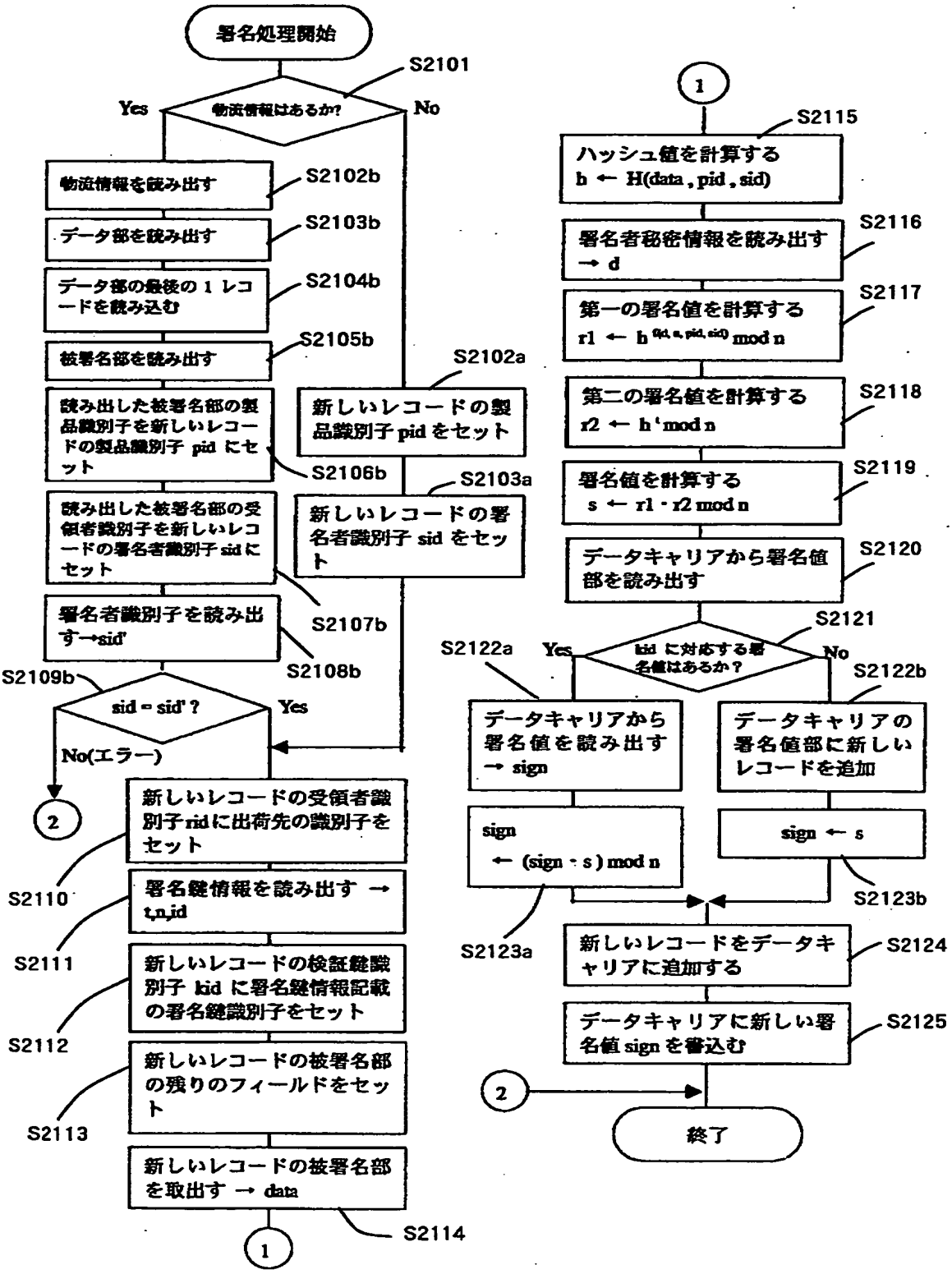
データキャリアに記憶される物流情報

【図２０】



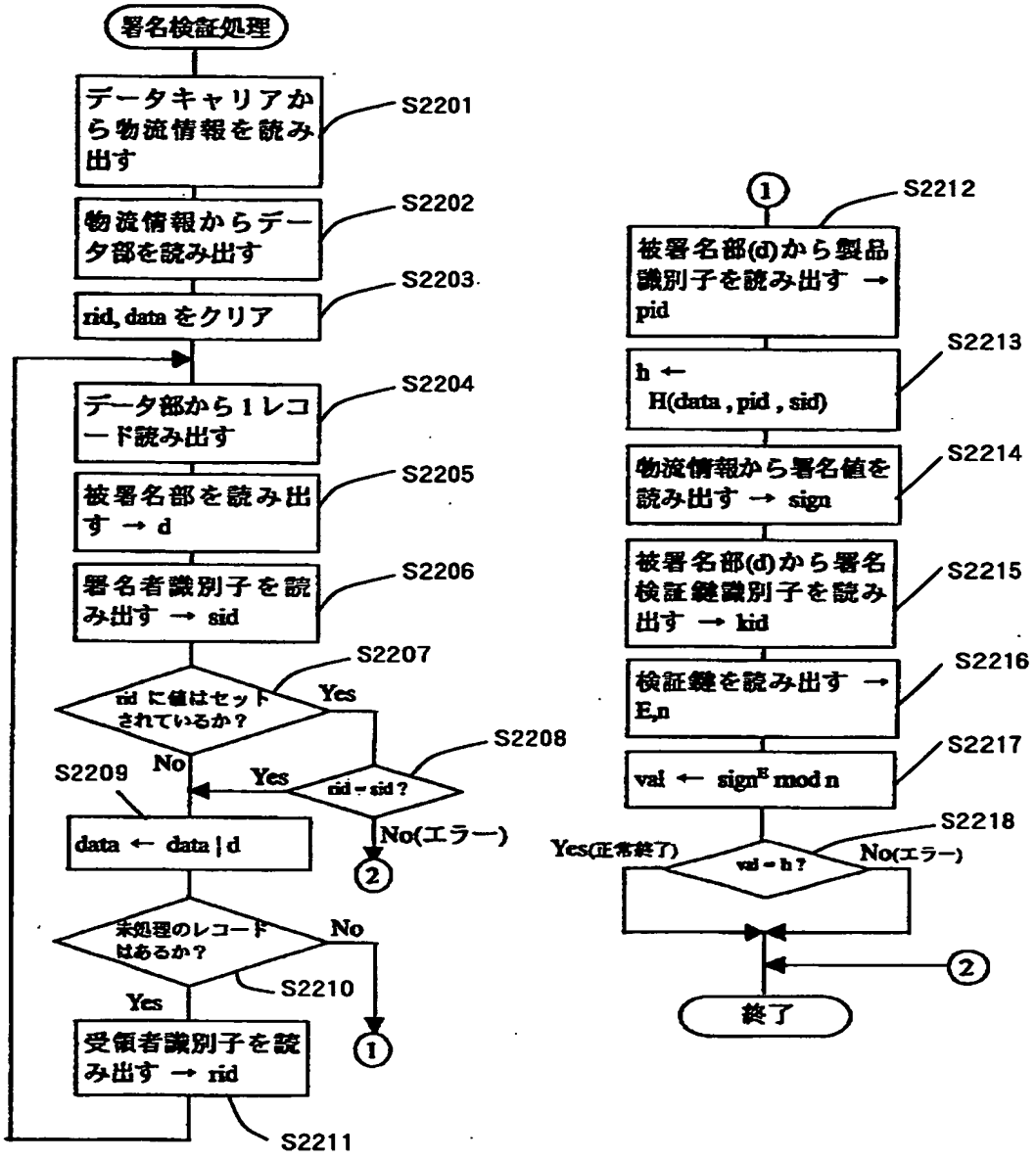
データキャリアに署名値を検証鍵毎に持つ場合の署名検証処理

【図２１】



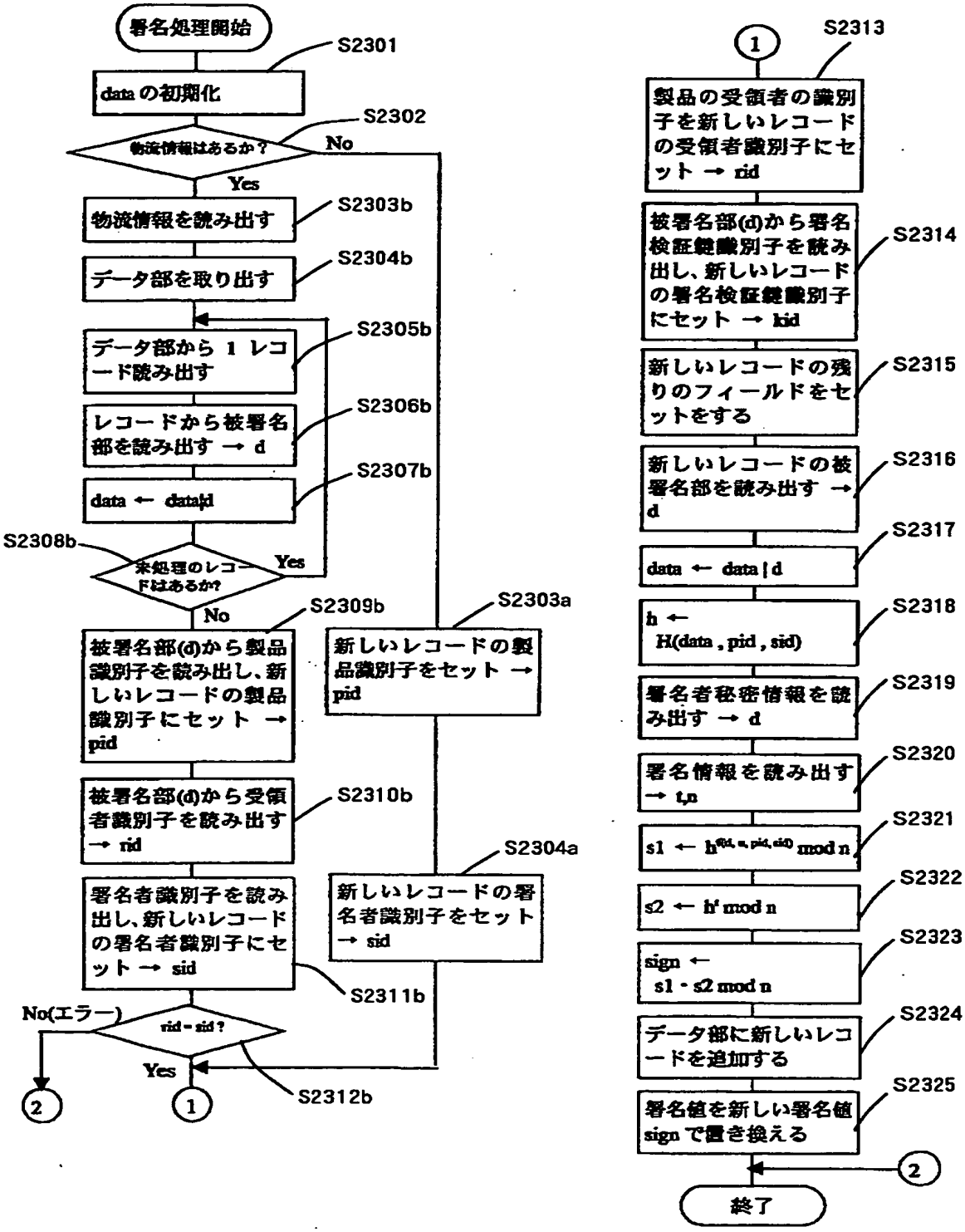
データキャリアに署名値を検証鍵毎に持つ場合の署名処理

【図２２】



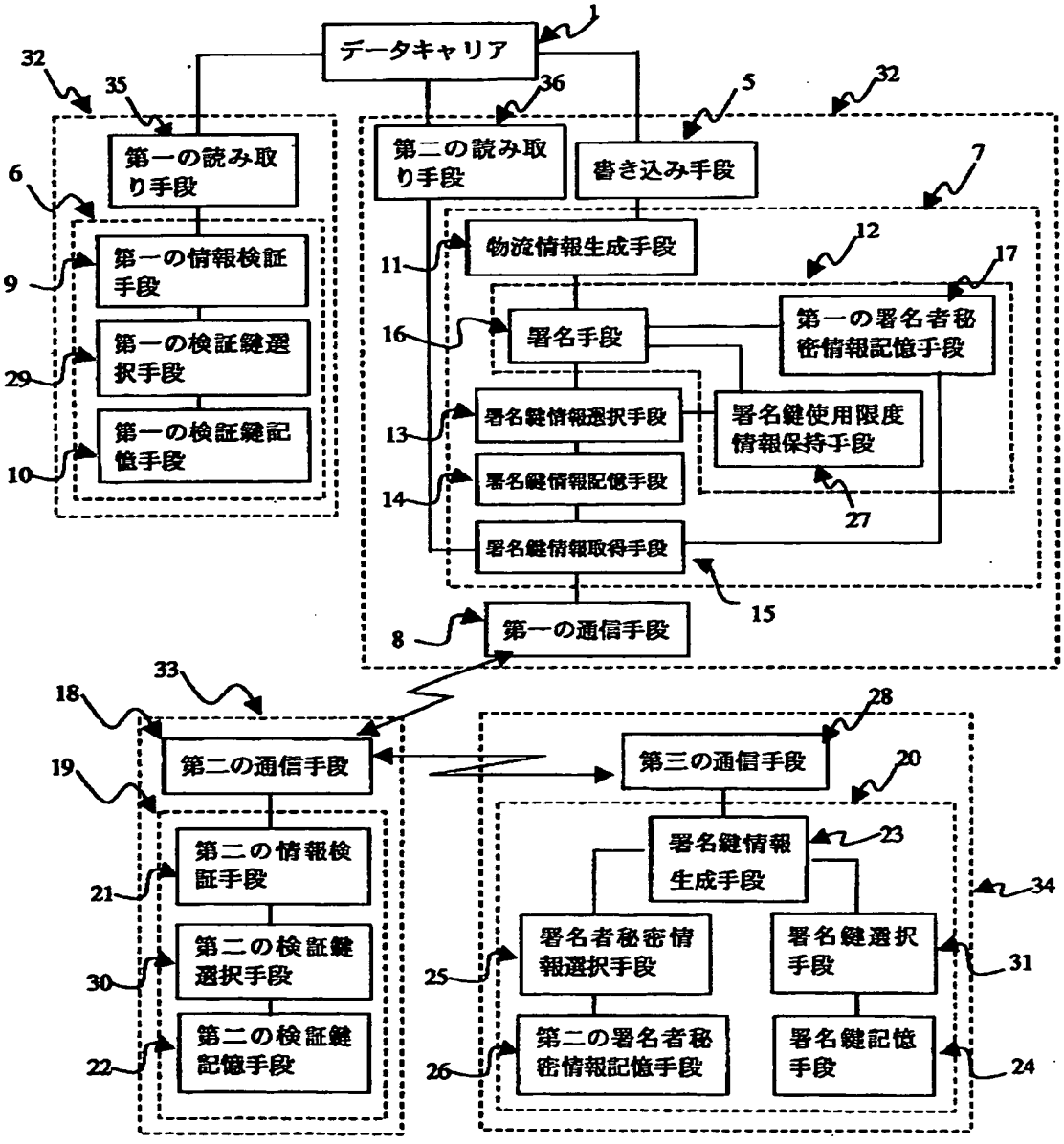
全体に署名をつける場合の検証処理（検証鍵は複数）

【図２３】



全体に署名をつける場合の署名処理 (検証鍵は複数)

【図２４】


データキャリア
1
32
35
第一の読み取り手段
6
第一の情報検証手段
9
第一の検証鍵選択手段
29
第一の検証鍵記憶手段
10
36
5
32
第二の読み取り手段
書き込み手段
7
物流情報生成手段
11
12
17
署名手段
16
第一の署名者秘密情報記憶手段
署名鍵情報選択手段
13
署名鍵使用限度情報保持手段
署名鍵情報記憶手段
14
27
署名鍵情報取得手段
15
8
第一の通信手段
33
18
第二の通信手段
19
28
第三の通信手段
20
第二の情報検証手段
21
署名鍵情報生成手段
23
34
第二の検証鍵選択手段
30
署名者秘密情報選択手段
25
署名鍵選択手段
31
第二の検証鍵記憶手段
22
第二の署名者秘密情報記憶手段
26
署名鍵記憶手段
24

変形例の構成

【書類名】　要約書

【要約】

【課題】　流通業者が、認証局から署名者ごとの証明書を取得する必要なしに、署名の検証を行えるようにする。

【解決手段】　署名手段１６は、物流情報に対してハッシュ値を計算し、変数ｈにセットする（ステップＳ４１２）。第一の署名者秘密情報記憶手段１７から署名者秘密情報を取出し変数ｄにセットする（ステップＳ４１３）。署名鍵情報選択手段１３は、署名鍵情報記憶手段１５から製品識別子ｐｉｄに対応する署名鍵情報を取出し、変数ｔ、ｎにセットする（ステップＳ４１４）。署名手段１６は、署名者秘密情報ｄを用い変数ｈに対して第一の署名値を計算し、変数ｒ１にセットする（ステップＳ４１５）。物流情報生成手段１１は、署名鍵情報ｔを用いて変数ｈに対して第二の署名値を次式で計算する（ステップＳ４１６）。計算結果ｒ１とｒ２を使って、最終的な変数ｈに対する署名値を計算する。

【選択図】　図４

出証特平１１－３０７４８２５

出　願　人　履　歴　情　報

識別番号　　〔０００００５４９６〕

１．変更年月日　　１９９６年　５月２９日
［変更理由］　住所変更
住　所　東京都港区赤坂二丁目１７番２２号
氏　名　富士ゼロックス株式会社